AD-A210 382

(2)

USER LANGUAGE CONSIDERATIONS IN MILITARY HUMAN-COMPUTER INTERFACE DESIGN

by

Daniel J. Pond

&

William K. Gabrenya

for

US Army Laboratory Command

June 30, 1988

Contract No. DAA03-86-D-0001
Delivery Order 0190
Scientific Services Program

DTIC
ELECTE
JUN 28 1989
S D
D &

The views, opinions, and/or findings contained in this report are those of the authors and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

Approved for public release;
distribution is unlimited.

89 6 28 003

CURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188
Exp. Date: Jun 30, 1986

a. REPORT SECURITY CLASSIFICATION	1b. RESTRICTIVE MARKINGS
a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
. PERFORMING ORGANIZATION REPORT NUMBER(S) Delivery Order 0190	5. MONITORING ORGANIZATION REPORT NUMBER(S) TCN 87-095
a. NAME OF PERFORMING ORGANIZATION Dr. Daniel J. Pond	6b. OFFICE SYMBOL (If applicable)
	7a. NAME OF MONITORING ORGANIZATION U.S. Army Research Office
c. ADDRESS (City, State, and ZIP Code) 11536 Indian River Drive Sebastian, FL 32958	7b. ADDRESS (City, State, and ZIP Code) P.O. Box 12211 Research Triangle Park, NC 27709-2211
a. NAME OF FUNDING/SPONSORING ORGANIZATION US Army Laboratory Command	8b. OFFICE SYMBOL (If applicable) SLCHE-CS
	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
c. ADDRESS (City, State, and ZIP Code) Ms. Maureen M. Larkin Human Engineering Laboratory Aberdeen Proving Ground, MD 21005-5001	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO

1. TITLE (Include Security Classification)

User Language Considerations in Military Human-Computer Interface Design (unclassified

2. PERSONAL AUTHOR(S)

Daniel J. Pond & William K. Gabrenya

13a. TYPE OF REPORT	13b. TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
FINAL REPORT	FROM _____ TO _____	1988, June, 30	112

6. SUPPLEMENTARY NOTATION Task was performed under a Scientific Services Agreement issued by Battelle, Research Triangle Park Office, 200 Park Drive, P.O. Box 12297, Research Triangle Park, NC 27709

COSATI CODES		
FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)

9. ABSTRACT (Continue on reverse if necessary and identify by block number)

This report details the soldier language/culture issues of possible relevance to US military effectiveness, especially in those systems with critical human-computer interfaces. In general, it was found that although few "applied" studies have been conducted, "basic" research as well as anecdotal reports indicate that such variables can, and frequently have, impaired military system effectiveness in terms of safety, morale, and/or mission accomplishment.

Because of this dearth of relevant applied research, a computer-based experimental task which incorporates US Army map symbology in a series of electronic map presentations has been developed as part of the current project. Although not yet refined to the level required of a research tool, descriptions of the experimental tasks, the maps, and the associated program code are included in this final project report.

0 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION
a. NAME OF RESPONSIBLE INDIVIDUAL	22b TELEPHONE (Include Area Code) / 22c OFFICE SYMBOL

FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

The authors wish express their gratitude for the support and guidance provided by Maureen M. Larkin and Larry A. Peterson of the US Army Human Engineering Laboratory. They also wish to acknowledge the contributions made throughout the term of this project by Dr. Ramon Mata of the Florida Institute of Technology .



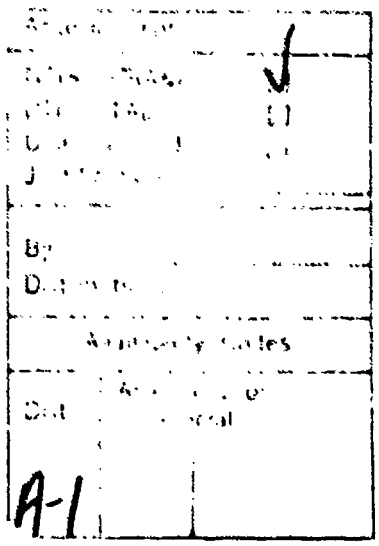


TABLE OF CONTENTS

INTRODUCTION	2
LANGUAGE: RELEVANCE FOR THE US MILITARY	2
Statistics	2
Occupational Issues	3
CONCEPTS AND ISSUES IN BILINGUALISM	4
Language Skills	4
Balanced versus Unbalanced Bilinguals	4
Processes Implicated in Performance Decrements	5
Individual Differences	6
Assessment of Language Ability	7
PERFORMANCE IMPLICATIONS OF BILINGUALISM	7
Stress Effects	7
Significance for the US Military	9
BILINGUALISM AND THE HUMAN-COMPUTER INTERFACE	11
Computer-specific Issues	11
The HCI Design Process	12
HCI Guidelines	12
A VARIABLY-INTENSIVE LANGUAGE TASK	14
REFERENCES	17
APPENDIX A	22
APPENDIX B	24
APPENDIX C	28
APPENDIX D	39
APPENDIX E	72

User language considerations in
military human-computer interface design

INTRODUCTION

This report details the soldier language/culture issues of possible relevance to US military effectiveness, especially in those systems with critical human-computer interfaces. In general, it was found that although few "applied" studies have been conducted, "basic" research as well as anecdotal reports indicate that such variables can, and frequently have, impaired military system effectiveness in terms of safety, morale, and/or mission accomplishment.

Because of this dearth of relevant applied research, a computer-based experimental task which incorporates US Army map symbology in a series of electronic map presentations has been developed as part of the current project. Although not yet refined to the level required of a research tool, descriptions of the experimental tasks, the maps, and the associated program code are included in this final project report.

LANGUAGE: RELEVANCE FOR THE US MILITARY

Statistics

Historically, "the United States has had a lot of experience recruiting aliens to its military, and at times the army has been 25 percent foreign-born" (Sinaiko, Curran, King, & Schneider, 1985). Major subcultural groups in the United States and in its armed services include Blacks, Hispanics, Asians, and Pacific Islanders. With a median age of 24 years, Hispanics as a group are six years younger than the general US population. In 1983, they numbered 15 million, which represents almost 7% of the population, and comprise the largest linguistically-distinct group and the second fastest growing group in the United States (Estrada, 1985).

These data, combined with the "baby bust" among White Americans, and the fact the limited English speaking (LES) Hispanics--most of whom are insular Puerto Ricans--tend to be well-educated and highly motivated, make Hispanics a desirable

recruiting pool (see, for example, Harman, 1984 and Holland, Rosenbaum, Stoddart, Redish, Harman, & Oxford-Carpenter 1984). Eitelberg (1985), who regards Hispanics "as an important manpower resource for the military", notes that while Hispanics are presently underrepresented in the military--comprising 1.5 % of the officer force and 3.9% of the enlisted force--they tend to be overrepresented in clerical and combat-arms specialties.

Occupational Issues

Deficits in language and basic skills among many Hispanics, particularly insular Puerto Ricans, may mean that they will not be able to use complex military hardware, particularly under the heavy workload conditions of combat. Although Chang & Lare (1985) reported no significant differences in low level text decoding and lexical knowledge abilities between English as a second language (ESL) and native English speaking (NES) recruits, the ESL subjects performed significantly worse on higher level tasks including text comprehension and information integration. Such performance differences undoubtedly contribute to the underrepresentation of ESL personnel in technical specialties requiring, for example, sufficient reading ability to comprehend long and complex technical manuals (see, e.g., Chang & Lare, 1985).

Triandis (1985) found that language difficulties limited the ability of Hispanic Naval personnel to understand their career options, leading them to become disaffected and unhappy. The Army has reported that some US soldiers born in Guam, the Philippines, Vietnam, and Puerto Rico, understand so little English that they are rendered "unreliable" (*Miami Herald*, May 22, 1986). In an extreme case, a Pakistani-born US sailor who failed out of a Naval electronics school due to language difficulties was charged with murdering one instructor and wounding two others (*Miami Herald*, November 17, 1986).

Salas, Kincaid, & Ashcroft (1980) found that 25% of the variance in drop-out rates from Naval recruit training could be predicted from English comprehension level. They further report that enlistment exam scores were not predictive of later communication difficulties, and recommend that remedial programs place emphasis on all four English language skills: reading, writing, speaking, and listening. Clearly, the imbalance between Hispanic work force availability and competence on language-intensive tasks shifts the emphasis for the armed services from recruitment to

training (Estrada, 1985; Harman, 1984). Unfortunately, however, Chipman (1985) notes that "researchers in the training area don't know enough about the effects of language proficiency, or about information processing under stress."

CONCEPTS AND ISSUES IN BILINGUALISM

Language skills

Full use of language involves development of four kinds of skill. Speaking and writing are called the language production or *encoding* skills, while listening and reading are termed language reception or *decoding* skills. As detailed by Macnamara (1967), four aspects of language may be distinguished within each of these four skills. These are: lexical (word), semantic (meaning), phonological (sound), and syntactic (structure) characteristics. Monolinguals may vary in their level of ability across each of these sixteen skill x aspect combinations, while a bilingual's abilities may vary across these sixteen combinations in both their native and secondary language(s). Degree of bilingualism, therefore, is a very complex and often inadequately-defined construct which may vary along numerous continua.

Diminished lexical, syntactical, or phonological competence may contribute to HCI problems. Research on optimal computer command names (see e.g., Rogers & Oborne, 1985) can identify words that are concrete, familiar, understandable, and high in imagery in one language--typically English--but these characteristics may be different for individuals for whom English is nondominant. Similarly, grammatical forms that are clear to native English speakers may be difficult for those who speak English as their second language. HCI guidelines or standards based exclusively on English text or developed with native English speaking subjects may, therefore, be invalid for ESL system operators.

Balanced Versus Unbalanced Bilinguals

Balanced bilinguals are individuals who have both good and equal ability in two or more languages, while unbalanced bilinguals are stronger in one language (or in one skill or aspect of language). Perfectly balanced bilinguals are rare because most bilinguals learn one language earlier and more thoroughly than the other(s) and therefore are, in some measure, deficient in their second language. Unbalanced bilingualism is not always apparent due to strategies

and compensatory processes employed by such individuals when communicating in the weaker language (Macnamara, 1967). These techniques include speech simplification and avoidance of hard-to-pronounce words.

Even among individuals who learn two languages simultaneously, or who thoroughly learn a second language, perfect balance is infrequently achieved because each of these languages may serve different communicative functions, such as domestic and social communications versus business or formal communications (Segalowitz, 1981). These different situations typically require different vocabulary, phraseology, syntax and/or level of literacy, and Macnamara (1967) has said that it is "pointless" to look for balanced bilinguals in circumstances having widely varying demands and characteristics. Such circumstances are said to involve *diglossia*, and bilinguals typically exhibit different proficiency levels in one or more of the language skills or aspects previously detailed. Estrada (1985) estimates that 50% of American Hispanics do not speak English in their homes which, of course, suggests that many Hispanics in the armed forces maintain different languages for different spheres of their lives.

Processes implicated in performance decrements

Psycholinguistic research on performance decrements frequently considers cognitive processing in bilinguals, and thus focuses on information storage, retrieval, and memory representations (Segalowitz, 1981). As detailed in later sections, spare capacity, workload, and stress effects are critical elements of these areas of inquiry.

Research on bilingualism has examined various types of interference effects and the contexts in which they occur (Hasselmo, 1967). Although there is not wide-spread agreement, three main categories of interference have been proposed:

1. lexical--interference involving the meaning of words;
2. grammatical--cross-language conflicts in syntax and word-combinations (morphology); and
3. phonological--over- or under-differentiation of sounds.

In HCI contexts, lexical interference could involve misinterpretation of English words that have a different meaning in the user's dominant language. Dornic (1977) claims that automatic

activation of words in the dominant language by the physical presence of their referents is one source of performance decrement among bilinguals. This could be viewed as lexical interference. Grammatical interference might occur when sentences are incorrectly interpreted because of misleading similarities in structure. Phonological interference leads to slower pronounceability of words, which slows verbal tasks and tasks that require covert speech. Such interference may be a more frequently occurring problem in the future as speech synthesis technology is increasingly incorporated into modern military and other systems.

Individual Differences

Although little research has been devoted to these variables (Shneiderman, 1987), a myriad of individual characteristics have been hypothesized to mediate the performance-user language relationship. A great potential for HCI research lies in this area if these many variables are organized in a theoretically meaningful manner, their relative strengths determined, and the practical ability of HCI designers to take account of them is assessed. Some examples of systematic attempts to incorporate attention to such variables into HCI design include IBM's development of the 1984 Olympic Message System (Boies, Gould, Levy, Richards, & Schoonard, 1985), and Apple Computer's guidelines for writing software for international markets (Apple Computer,1987, 1986).

Because these individual difference variables have, by-and-large, not been empirically assessed in military- or HCI-relevant situations, they are merely listed here without further expansion.

Van Der Veer, Tauber, Waerns, & Muylwijk (1985):

-field dependence/independence,
-impulsivity/reflectivity,
-operation learning/comprehension learning,
-introversion/extraversion,
-negative fear of failure,
-perception of own competence,
-knowledge and experience,
-intelligence.

Shneiderman (1987):

-Myers-Briggs Type Inventory types.

Coovert & Goldstein (1980):

-internal/external locus of control.

The following are also hypothesized to influence performance by bilinguals when working in their subdominant language:

-psychological differentiation,
-gross effects of ecology,
-individualism (social field dependence, social facilitation, social loafing, social interaction styles, intercultural interaction),
-delay of gratification,
-emotionality,
-values (materialism, power distance, uncertainty avoidance).

Assessment of language ability.

A variety of instruments have been developed to assess English competence, and these measures show moderate to high levels of covariance (e.g., see Salas, et al., 1980). Degree of bilingualism has been estimated by comparing several types of language competencies (i.e., encoding versus decoding; writing versus speaking) in the languages used by the respondent (although recall that to be thorough, this must be done for sixteen skill x aspect combinations). Interrelationships among bilingualism measures have been reported to range widely (Fishman & Cooper, 1971). English language measures pick up relatively fine, quantifiable degrees of English competency, whereas bilingualism measures assess extent and direction of imbalance.

Although complex and sophisticated methods have been developed to measure bilingualism, research seems to indicate that the best predictors of language dominance are the language used in the home and proficiency self-ratings. Macnamara (1967) warns, however, that bilingual proficiency is best measured on the skills that are of interest. This position is consistent with the observation that language proficiencies are contextually delimited. It also suggests that language abilities in the military should be assessed in relevant contexts, including HCIs and stressful situations.

PERFORMANCE IMPLICATIONS OF BILINGUALISM

Stress effects

Although imbalances in language proficiency may, as discussed, be masked through a number of techniques, these are

likely to become manifest under stress (Dornic, 1980). Recently, for example, a Hialeah, Florida Hispanic police officer was forced to resign his position because "...when he got excited, he would revert to Spanish" (*Miami Herald*, April 18, 1988). He was, therefore, unable to communicate effectively, which rendered him unable to perform his duties satisfactorily . We may interpret such behavior in light of *Drive Theory* (see, e.g., Spence, 1958; Hull, 1952) which holds that the **E**xcitatory potential for responses under high levels of **D**rive are those having the greatest **H**abit strength--that is, E = D x H. In unbalanced bilinguals, therefore, the language used (i.e., the response with the greatest excitatory potential) under stressful conditions (which increases Drive) is that which is stronger or predominant (i.e., has greater habit strength).

In a series of reports, Dornic and his associates have postulated that greater effort and cognitive processing resources are required by individuals working in their nondominant language (see, for example, Dornic & Wirberg, 1983; Dornic & Dornic, 1981; Dornic, 1979). Hispanic ESL soldiers working with high technology, computer-based military systems (and the associated esoteric terminology) appear particularly susceptible to the adverse performance effects associated with diglossia. Such performance decrements may be exacerbated during military operations which may expose soldiers to a variety of stressors including environmental (noise, temperature), task information load (complexity, input rate, unexpected events), emotion (incentive, fear, risk-taking), and mental fatigue (Dornic & Dornic, 1981; Dornic, 1977).

Such effects may be interpreted through Kahneman's (1973) concept of "spare capacity" for processing information. As a situation becomes more demanding--by, for example, an increase in task complexity, presence of environmental stressors, or use of nondominant language, this spare capacity is tapped in order to maintain a satisfactory level of proficiency. As these demands approach, and then exceed, an individual's processing capacity, performance deteriorates

If required to perform two tasks simultaneously under demanding conditions, an operator will typically try to maintain satisfactory performance on the "primary" or more critical task at the expense of poorer performance on the less important, "secondary" task. *Secondary task methodology*, therefore, focuses

upon performance impairments on these secondary tasks as the most sensitive indicator of operator workload and the level at which spare capacity is depleted. Stress-induced performance decrements in bilinguals have been found through use of secondary task methodology and corroborated via reports of perceived effort (Dornic, 1980).

Several hypotheses regarding the observed relationship between stress and nondominant language performance decrements have been proposed. Noise appears to mask inner speech, and this induces more problems when an individual is using the nondominant rather than the dominant language. Further, since tasks performed in a nondominant language are by definition more complex, performance-arousal effects may be particularly problematic. Since many stressors increase arousal, nondominant language tasks are often performed at greater than optimal arousal levels, resulting in performance decrements. Finally, there are some indications that "mental fatigue" reduces the ability of bilinguals to keep their language systems distinct and leads to less effective memory search and impaired short-term memory (see Dornic, 1980).

Significance for the US military

A number of reports indicate that US military operations have been adversely affected by soldier language problems. During a workshop devoted to *Hispanic subpopulations and Naval service*, for example, an incident was described in which a Marine tactical air exercise was curtailed because some of the Hispanic personnel reverted to speaking Spanish while under the stress of military operations (Williams, 1985). The paucity of knowledge regarding the effects of language proficiency or information processing under such stress has led a representative of the Office of Naval Research to conclude that "...it is questionable whether Hispanics should be required to be able to write in a second language [i.e., English] while under the stresses of combat" (Chipman, 1985).

There is some evidence that bilingualism is a greater problem in the military than are literacy levels among high-school educated monolinguals. For example, Chang and Lare (1985) claim that text comprehension of bilinguals is worse than that of marginally-literate NES recruits and, of significance for computer-based systems and HCIs, this difference held true for both auditory and visual presentations of text.

Research on performance by bilinguals indicates that both encoding and decoding is slower in the nondominant language. For example, words can be matched to pictures (encoding) faster in the dominant than the nondominant language (Dornic, 1977) which may contribute to the reported language/culture differences in use of symbolic information (see Pond, in press; Apple Computers, 1986). Dornic (1980) suggests that "where fast decoding of verbal signals is vital, information should always be given in the operator's dominant language."

Potter, So, Von Eckardt, & Feldman (1984) delineate two hypotheses regarding the association between equivalent words in a bilingual's two languages. The *word association* hypothesis proposes a direct connection between words in the two languages. In order to name an object in one's nondominant language, for example, the relevant word must first be retrieved in the dominant language, and then a translation of this word made to the weaker language. On the other hand, the *concept mediation* hypothesis proposes that both languages have direct access to the object/concept stimulus. Although the word association hypothesis is said to have some intuitive appeal to language learners, Potter, et al. (1985) found that both novice and expert bilingual subjects perform in a manner predicted by the concept mediation hypothesis. These findings, if substantiated, may mitigate somewhat the potential performance decrements hypothesized here for ESL soldiers because no additional response latency would be incurred as "internal translations" between the subdominant and dominant languages are required before stimulus processing can begin.

Soldier native language issues may also adversely impact the effectiveness of multi-national military operations. Sidorsky, Geilman, & Moses (1979), for example, refer to such problems within the North Atlantic Treaty Organization, and Simpson (1980) notes the potential lethality of "improper communications" during joint US Army-Republic of Korea Army operations.

Such difficulties may arise in soldier-system interchanges as well as in interpersonal communications. In 1979, Parrish, Gates, & Munger reported that the US Army had sixty computer-based systems under development. Given the resulting increased incidence of human-computer interfaces (HCIs) within the Army, we may anticipate an increase in language-induced operational problems because such interactions are often language-based. Indeed, as

noted above, language/culture differences can impair HCI effectiveness in graphic-based systems as well. Bersh, Moses, & Maisano (1978), for example, note that even a symbol as intuitively clear as a directional arrow may be misinterpreted because of observer cultural influences.

BILINGUALISM AND THE HUMAN-COMPUTER INTERFACE

Computer-specific issues

An interesting and important issue in HCI research is how computer tasks differ from their counterparts performed on paper. This issue is relevant to the applicability of existing culture/linguistic and task performance literature to HCI design. A number of studies have found that certain tasks are performed faster and better on paper than on computers. Wright and Lickorish (1983) found that speed and accuracy of proofreading are superior on paper. Gould and Grischkowsky (1984) found superior proofreading performance on paper than on a CRT, but did not find differences in self-reported affect, comfort, or visual problems. Gould (1986) found that the paper-CRT difference could be eliminated by using antialiased screen characters (characters formed by pixels of variable brightness, producing better resolution) and fonts such as those used in printing.

The superiority of paper over all but the best CRT displays suggests that most existing computer hardware places a burden on performance not present in traditional paper-and-pencil tasks, thus increasing task complexity. It seems likely that care in HCI design can obviate much of this disadvantage of CRTs by, for example, following Smith and Mosier's (1984) screen display guidelines and, of course, effectively accounting for variations in user language capabilities.

Several other characteristics of computers can be hypothesized to set them apart from some non-computer-based information handling. These include:

- controllability--Many paper-and-pencil and verbal tasks are under user control. In much computer software the computer is "in control", and it only accepts specific user inputs at specific points in time.
- predictability--Computers can be unpredictable to less-experienced users. Some tasks, such as data entry and display,

word processing, and communication are inherently less predictable on a computer than they are in their non-computer forms.

• familiarity.--For many inexperienced users, the computer is a completely unfamiliar and alien device. This may be particulary true for individuals from lower income families (who, therefore, could not purchase a home computer), and those with fewer years of education (and, thus, have less possible exposure time with computers)..

It may again be anticipated that these variables would be more critical for the performance of ESL soldiers due to the higher workload levels inherent in any English language-based operation. Acceptance of, and attitudes towards computers may have implications for training and performance which are at least as important as HCI design issues. It may also prove true that poor HCI design engenders negative attitudes among new users, leading to motivational and performance decrements.

The HCI design process

Gould & Lewis (1985) have developed an HCI design process that includes empirical testing with end-users and which contains three design principles. These are:

- an early focus on users and tasks;
- use of empirical measurement in the design process; and
- incorporation of an iterative approach to the design.

Such an approach appears amenable to consideration of user language capabilities, individual differences, and a military system context as recommended in this report.

HCI guidelines

Formal guidelines (e.g., Smith & Mosier, 1984), military standards (e.g., MIL-STD 1472C), and HCI texts (e.g., Shneiderman, 1987) are attempts to organize existing knowledge about HCI design and to influence software system implementations. Gould and Lewis (1985) suggest that guidelines should be viewed as helpful in getting started and in building prototypes, but we must recognize that they are limited in their ability to guide HCI design in specific contexts. Given this context-dependence of HCI design, Gould (1987) maintains that guidelines and standards cannot take the place of behavioral work. Indeed, even the author of a guideline document (Smith,1986) warns that present knowledge does not warrant the imposition of standards, but indicates that standards may eventually evolve through empirical testing and implementation of guidelines.

Based on the present review, it appears imperative that user language/culture variables be included in this pre-standard development testing.

As in many design approaches which adequately consider *human factors*, Smith & Mosier (1984) frequently cite "flexibility" as an important design concept (e.g., see the section introductions for Data Entry, Data Display, Sequence Control, and User Guidance). To effectively aid software designers in providing appropriate "flex" for ESL users, however, requires more specific, empirically derived recommendations. Only in rare instances are language/culture variations specified in the current guidelines--for example, in ¶ 2.4-29, Conventional Assignment of Color Codes, Smith & Mosier indicate the "red is associated with danger (in our society)."

Examination of Smith and Mosier (1984) reveals a significant number of additional guidelines that may have implications for writing software for bilingual end-users and that are, therefore, candidates for empirical investigations focussing on in ESL soldiers in combat stress situations. These include:

¶ 1.0-7: User paced data entry may compensate for slower processing in nondominant languages.

¶ 1.0-14: Short data items ease demands on short term memory.

¶ 1.3-5: Natural units of text may or may not be "natural" given grammatical interference effects (see above).

¶ 2.0-3: Data displayed in usable form may have implications, for example, for presentation of data in feet versus meters.

¶ 2.0-11: Familiar wording and verb forms, if appropriate for the user's nondominant language, may promote comprehension.

¶ 2.0-14: Minimal use of abbreviations may be particularly helpful for ESL soldiers whose nondominant language referents will not match an English-based acronym.

¶ 2.1.1-7: Conventional punctuation may not be the same for all languages (see, e.g., Apple Computers, 1986).

¶ 2.1.1-9: Use of sentences beginning with main topic may not be appropriate in all languages.

¶ 2.1.1-14: Use of active voice may not be appropriate for all languages.

¶ 2.1.4-4: Standardized graphic symbology may not be suitable for all cultures (see, e.g., Pond, in press; Apple Computers, 1986).

¶ 2.1.2; 2.3: If cultural differences in field dependence prove sufficiently robust, the structure of data forms as well as display formatting may prove especially critical for some minorities.
¶ 2.4-29: As noted by Smith & Mosier, conventional color codes may lead to a color interference effect across cultures.
¶ 3.0-13: User expectations regarding control-display compatibility may vary across cultures.
¶ 3.1.7-1: Implementation of effective constrained natural languages may prove difficult for bilinguals.
¶ 3.1.8-1: Graphical interaction in dialogs may be useful for bilinguals, although icons are known to differ across cultures (Pond, in press; Apple Computers, 1986;).

A VARIABLY-INTENSIVE LANGUAGE TASK

Because, as indicated, there has been virtually no applied research related to user language effects on military performance, an experimental task was devised for future use as part of the current project. Although it was not possible to fully refine and validate the task as part of this contract, the task is fully operational, and its parameters believed to be largely appropriate for use.

This task enables the experimenter to vary the language intensiveness of the experimental instructions (called "prompts" or "scripts" depending on the condition) from none through severe. The task is Macintosh computer-based, uses six military map symbols from the US Army Field Manual FM 21-30 as target stimuli (see Appendix A), and employs electronic representations of maps as backgrounds. Appendix B contains three sample task presentations, and Appendix C depicts each of the ten map backgrounds. Note, however, that due to printer limitations, these hardcopy representations are considerably inferior.in quality to their electronic counterparts used for research.

In all conditions, the subject is to locate a particular symbol, the English and native language meanings of which have been previously memorized. In all cases, target symbol detections are indicated via a trackball/pushbutton input device. Response latencies, numbers of hits, and misses are recorded as dependent measures.

For the first three conditions, one target symbol and fifteen "distractor" symbols are presented on each map (see Appendix B,). These task conditions are as follows:

1. The "prompt" is simply another representation of the symbol itself. The subject is to locate the specified target symbol on the map and indicate its position. This condition, of course, requires no language processing.
2. The (hypothesized) easiest language processing condition is one in which the target symbol prompt is delivered as the corresponding word in the subject's native language (including English as appropriate).
3. At the (again, hypothesized) next level of language intensiveness, ESL subjects are prompted with the corresponding word in English.

For the remaining conditions, three symbols of each type are presented on each map--i.e., eighteen total per map presentation There are one correct and two incorrect symbols of the specified target type. The correct target is found by following the navigational instructions in the stimulus "script"; A sample map/script is presented Appendix B. These "Orienteering" task conditions are as follows:

4. Navigational instructions are presented in the subject's native language (including English, as appropriate).
5. Navigational instructions are presented in "good" English--i.e., prose constructed according to Smith & Mosier (1984) guidelines.
6. Navigational instructions are presented in "degraded" English--i.e., prose constructed in violation of one or more Smith & Mosier (1984) guideline.

It is intended that each of the above conditions be more difficult than that which precedes it due to the language intensiveness of the stimuli--except for NES subjects for whom condition 2 is equivalent to condition 3 and condition 5 equivalent to condition 6.

Sample scripts in "good English", "degraded English", Spanish, and Korean are presented. in Appendix D. For each trial detailed therein is a three digit code. The first digit, which ranges between 1 and 10, indicates the number of the map which serves as the background. The second digit, which varies from 1 to 6, indicates the symbol (see Appendix A) which is to be the target on that trial,

and the third digit, which also ranges between 1 and 10, specifies the language conditions. These are:

1. Good English script
2. Degraded English script
3. English script for native English speaking subjects (same text as used in 1, above)
4. Spanish script for native Spanish speaking subjects
5. Korean script for native Korean speaking subjects
6. English prompt
7. English prompt for native English speaking subjects (same words as used in 6, above)
8. Spanish prompt for native Spanish speaking subjects
9. Korean prompt for native Korean speaking subjects
10. target symbol iconic prompt

Appendix E contains the *Macintosh Pascal* program source code which was compiled with a *Lightspeed Pascal* compiler. This code references Invention Software's *Programmer's Extenders*, Volumes 1 and 2 which are required to run this program.

REFERENCES

Apple Computer (1986). Human interface guidelines. Cupertino, CA: Apple Computer, Inc.

Apple Computer, The Human Interface Group (1987). HyperCard stack design guidelines. Cupertino, CA: Apple Computer, Inc.

Bersh, P., Moses, F.L., & Maisano, R.E. (1978). Investigation of strength of association between graphic symbology and military information (Technical Paper 324). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

Boies, S. J., Gould, J. D., Levy, S., Richards, J. T., & Schoonard, J. (1985). The 1984 Olympic message system--A case study in system design (Report No. RC 11138 [#50065]). Yorktown Heights, NY: IBM Research Center.

Chang, F. R., & Lare, C. M. (1985). Text comprehension processes in bilinguals (Report No. NPRDC TR 85-34). San Diego, CA: Office of Naval Research.

Chipman, S. (1985). Discussion: English language. In H.W. Sinaiko, P.M. Curran, B.T. King, & J.M. Schneider, (Eds.). Hispanic subpopulations and naval service. (Report No. SI/MRAS/TR-11). Arlington, VA: Office of Naval Research.

Coovert, M.D. & Goldstein, M. (1980). Locus of control as a predictor of users' attitude toward computers. Psychological Reports, 47, 1167-1173.

Department of Defense. (1981). Human engineering design criteria for military systems, equipment, and facilities (MIL-STD-1472C). Washington, DC: U.S. Government Printing Office.

Department of the Army. (1970). Field Manual, Military Symbols (FM 21-30). Washington, DC: U.S. Government Printing Office.

Dornic, S. (1977). Information processing and bilingualism Report number 510). Stockholm, Sweden: Department of Psychology, University of Stockholm.

Dornic, S. (1979). Information processing in bilinguals: Some selected issues. Psychological Research, 40, 329-348.

Dornic, S. (1980). Language dominance, spare capacity and perceived effort in bilinguals. Ergonomics, 23, 369-377.

Dornic, S., & Dornic, V. (1981). Perceived effort as a measure of mental load in nonbalanced bilinguals. Le Travail humain, 44, 251-259.

Dornic, S. & Wirberg, A. (1983). Note on intralingual and interlingual interference in the Stroop task (Report number 611). Stockholm, Sweden: Department of Psychology, University of Stockholm.

Eitelberg, M.J. (1985). Historical data on Hispanics in the US Military. In H.W. Sinaiko, P.M. Curran, B.T. King, & J.M. Schneider, (Eds.). Hispanic subpopulations and naval service. (Report No. SI/MRAS/TR-11). Arlington, VA: Office of Naval Research.

Estrada, L. (1985). Demographic overview of Hispanic America. In H.W. Sinaiko, P.M. Curran, B.T. King, & J.M. Schneider, (Eds.). Hispanic subpopulations and naval service. (Report No. SI/MRAS/TR-11). Arlington, VA: Office of Naval Research.

Fishman, J., & Cooper, R. (1971). The interrelationships and utility of alternative bilingualism measures. In W. Whiteley (Ed.), Language use and social change. London: Oxford University Press.

Gould, J. D. (1986). Reading is slower from CRT displays than from paper: Some experiments that fail to explain why (Report No. RC 11709). Yorktown Heights, NY: IBM Thomas J. Watson Research Center.

Gould, J. D., & Grischkowsky, N. (1984). Doing the same work with hard copy and with cathode-ray tube (CRT) computer terminals. Human Factors, 26, 323-337.

Gould, J., D. & Lewis, C. H. (1985). Designing for usability--key principles and what designers think. Communications of the ACM, 28, 300-311.

Harman, J. (1984). Three years of evaluation of the Army's basic skills education program. (ARI Report No. 1380). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

Hasselmo, N. (1969). How can we measure the effects which one language may have on the other in the speech of bilinguals? In L. G. Kelly (Ed.), Description and measurement of bilingualism: An international seminar. Toronto: University of Toronto Press.

Holland, V. M., Rosenbaum, H., Stoddart, S., Redish, C., Harman, J., & Oxford-Carpenter, R. L. (1984). English-as-a-second-language programs in basic skills education program I (Report No. 1359). Alexandria, VA: U. S. Army Research Institute for the Behavioral and Social Sciences.

Hull, C.L. (1952). A behavior system. New Haven, CT: Yale University Press.

Kahneman, D. (1973). Attention and effort. Englewood Cliffs, NJ: Prentice-Hall.

Macnamara, J. (1967). The bilingual's linguistic performance--A psychological overview. Journal of Social Issues, 23(2), 58-77.

Parrish, R.N., Gates, J.L., & Munger, S.J. (1981). Design guidelines and criteria for user/operator transactions with battlefield automated systems. Vol. 1: Executive summary (Research report 1320). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

Pond, D.J. (in press). Military symbologies: Overview and select annotated bibliography. Aberdeen, MD: US Army Human Engineering Laboratory.

Potter, M.C., So, K-F., Von Eckardt, B., & Feldman, L.B. (1984). Lexical and conceptual representation in beginning and proficient bilinguals. Journal of Verbal Learning and Verbal Behavior, 23, 23-38.

Rogers, Y, & Oborne, D. J. (1985). Some psychological attributes of potential computer command names. Behavior and Information Technology, 4, 349-365.

Salas, E., Kincaid, J. P., & Ashcroft, N. (1980). An assessment of Hispanic recruits who speak English as a second language (TAEG Report No. 86). Orlando, FL: Naval Training Equipment Center.

Segalowitz, N. S. (1981). Issues in the cross-cultural study of bilingual development. In H. C. Triandis & A. Heron (Eds.), Handbook of cross-cultural psychology Volume 4: Developmental Psychology. Boston: Allyn and Bacon.

Shneiderman, B. (1987). Designing the user interface: Strategies for effective human-computer interaction. Reading, MA: Addison-Wesley.

Sidorsky, R.C., Gellman, L.H., & Moses, F.L. (1979). Survey of current developments in tactical symbology: Status and critical issues (Working Paper 79-03O). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

Simpson, J.M. (1980). Wartime interoperability problems posed by differences in South Korean and United States Army tactics. Fort Leavenworth, KS: U.S. Army Command and General Staff College.

Sinaiko, H. W., Curran, P. M., King, B. T., & Schneider, J. M. (Eds.) (1985). Hispanic subpopulations and naval service. (Report No. SI/MRAS/TR-11). Arlington, VA: Office of Naval Research.

Smith, S.L. & Mosier, J.N. (1984). Design guidelines for user-system interface software (ESD-TR-84-190). Hanscom Air Force Base, MA: US Air Force, Air Force Systems Command.

Spence, K.W. (1958). A theory of emotionally based drive (D) and its relation to performance in simple learning situations. American Psychologist, 13, 131-141.

Triandis, H. C. (1985). Cultural patterns of Hispanics vis-a-vis military service. In H.W. Sinaiko, P.M. Curran, B.T. King, & J.M. Schneider, (Eds.). Hispanic subpopulations and naval service. (Report No. SI/MRAS/TR-11). Arlington, VA: Office of Naval Research.

Van Der Veer, G. C., Tauber, M. J., Waerns, Y., Muylwijk, B. V. (1985). On the interaction between system and user characteristics. *Behavior and Information Technology, 4*, 289-308.

Williams, A.C. (1985). Discussion: English language. In H.W. Sinaiko, P.M. Curran, B.T. King, & J.M. Schneider, (Eds.). Hispanic subpopulations and naval service. (Report No. SI/MRAS/TR-11). Arlington, VA: Office of Naval Research.

Wright, P., & Lickorish, A. (1983). Proof-reading texts on screen and paper. Behavior and Information Technology, 2(3), 227-235.

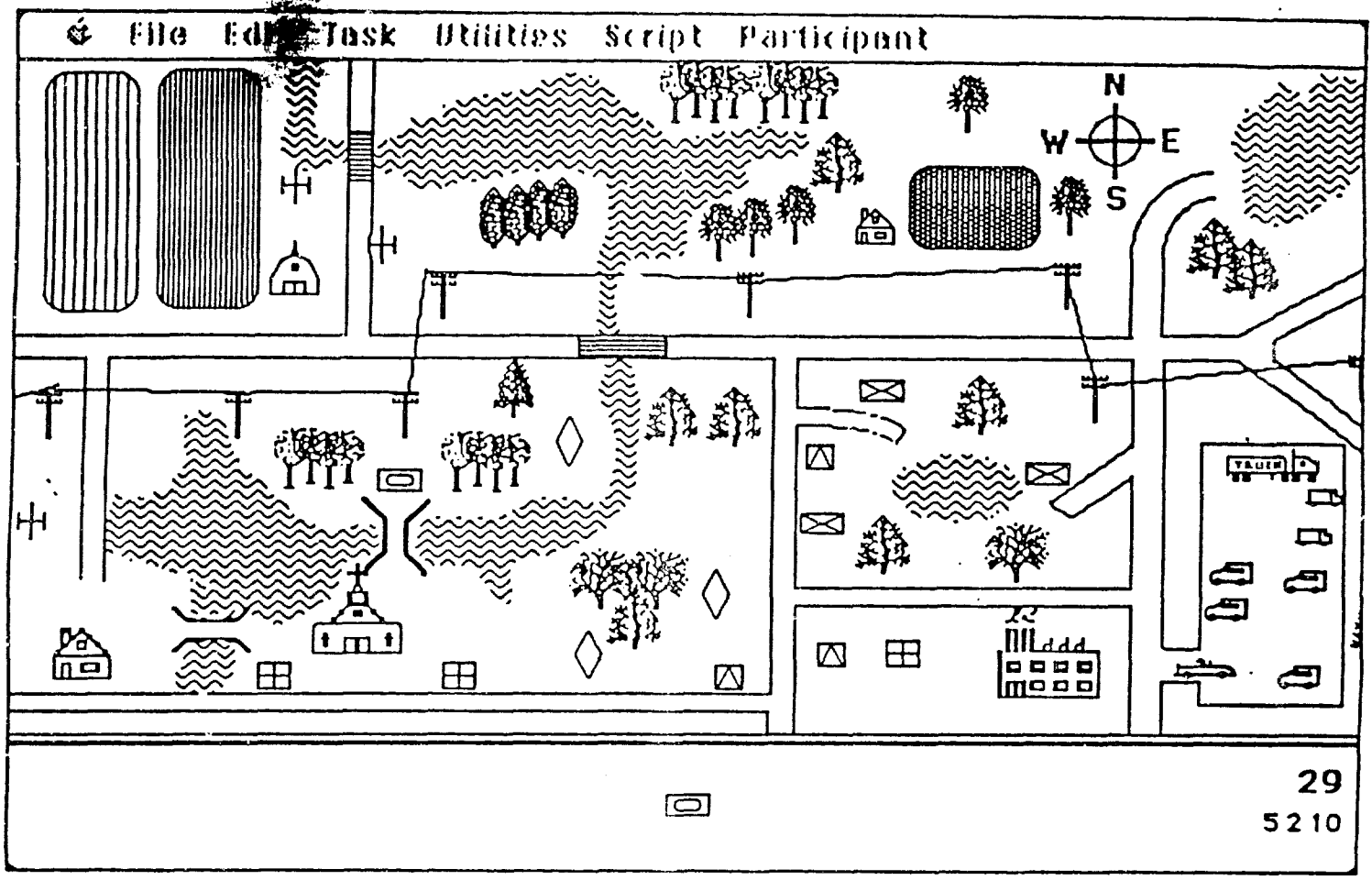
APPENDIX A

MILITARY MAP
TARGET SYMBOLS

Symbol	English	Korean	Spanish
START	start	--	--
	Infantry	보병부대	Infanteria
	Tank	탱크	Tanque
	Gun	대포	Cañon
	Anti-tank gun	탱크 저항용 대포	Cañon anti-tanque
	Hospital	병원	Hospital
	Personnel carrier	전령	Vehiculo de transporte de personal

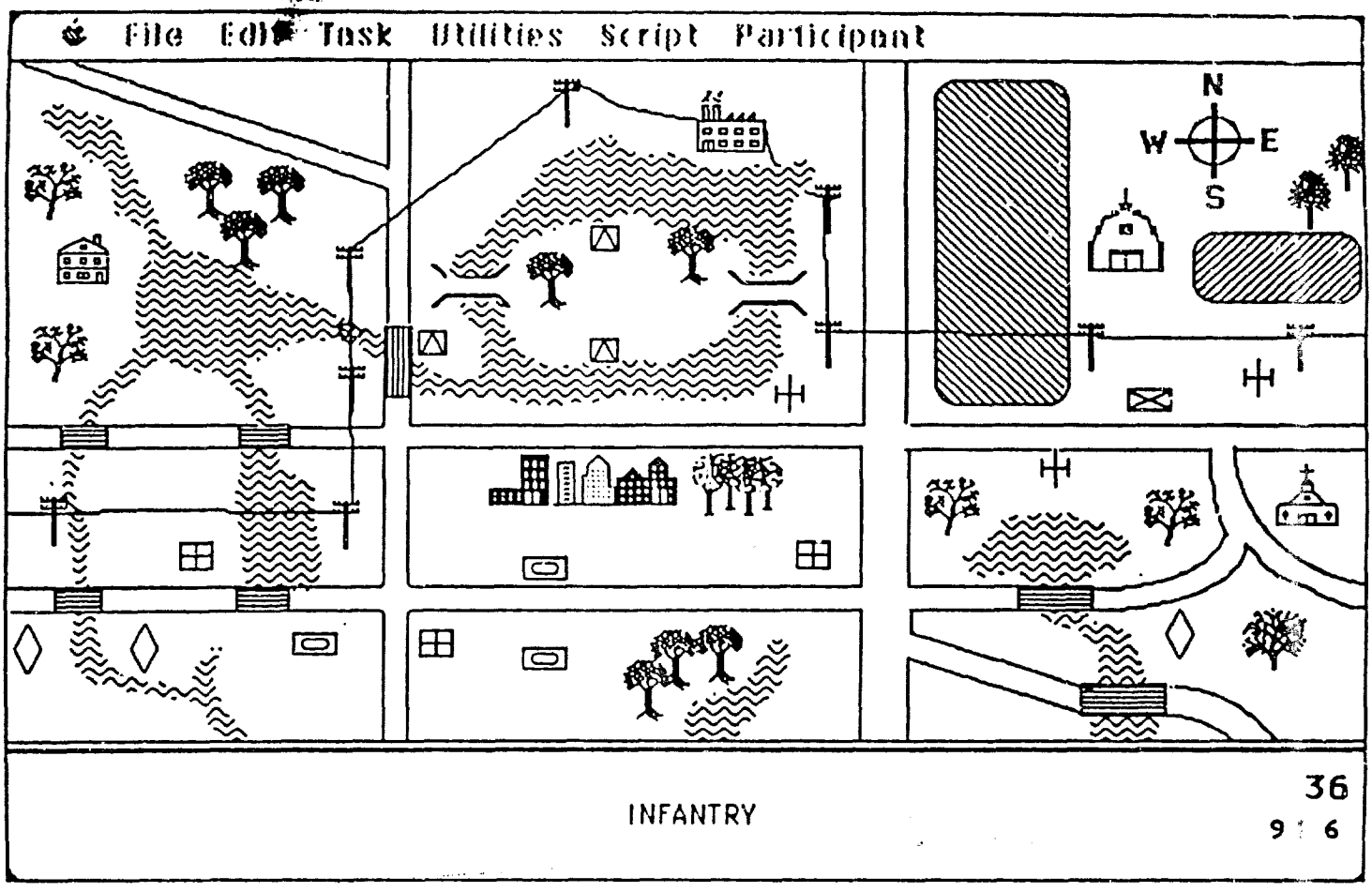
APPENDIX B

SAMPLE TASK STIMULI



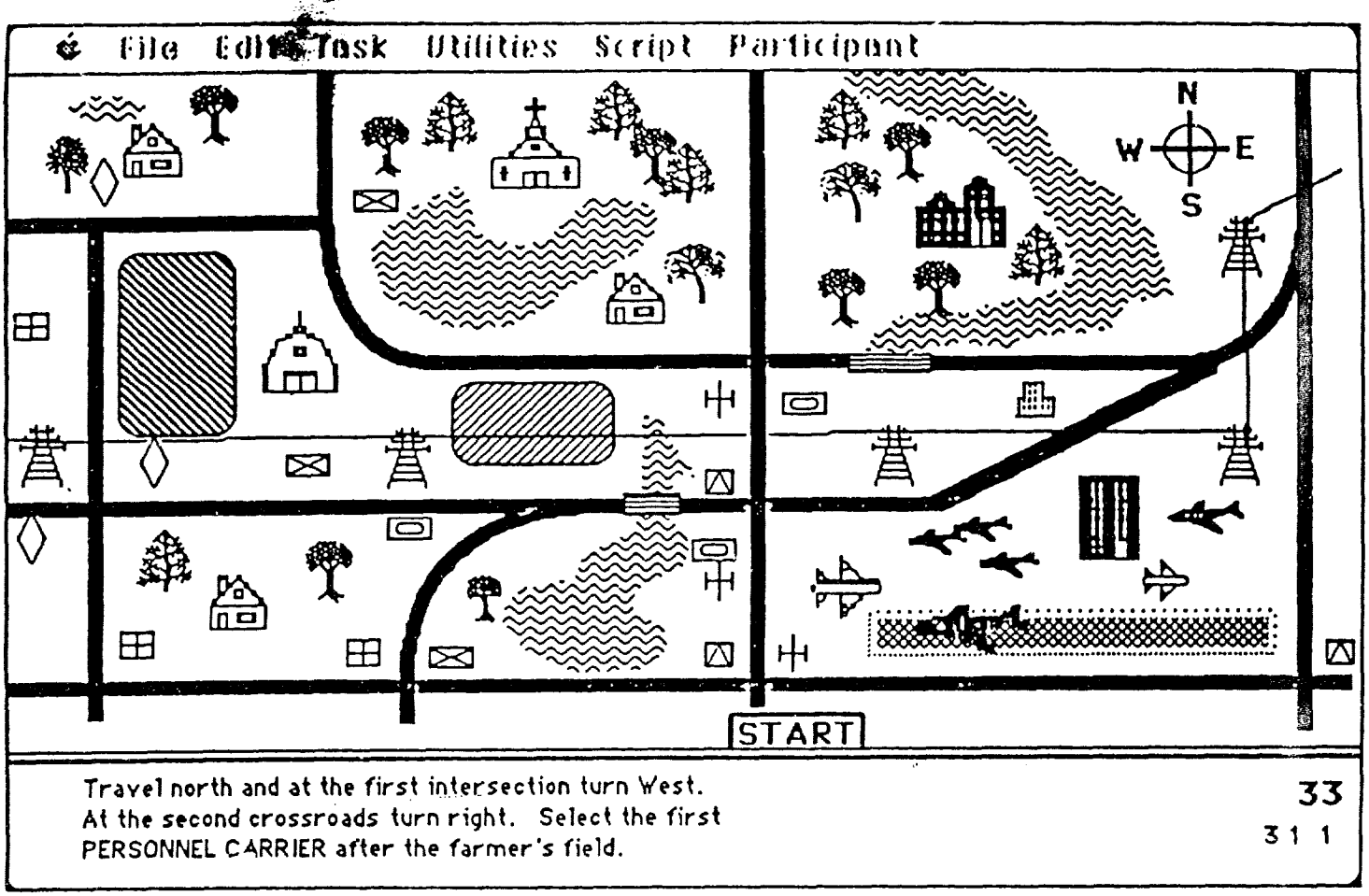
ICON PROMPT

Best Available Copy



ENGLISH WORD PROMPT

Best Available Copy

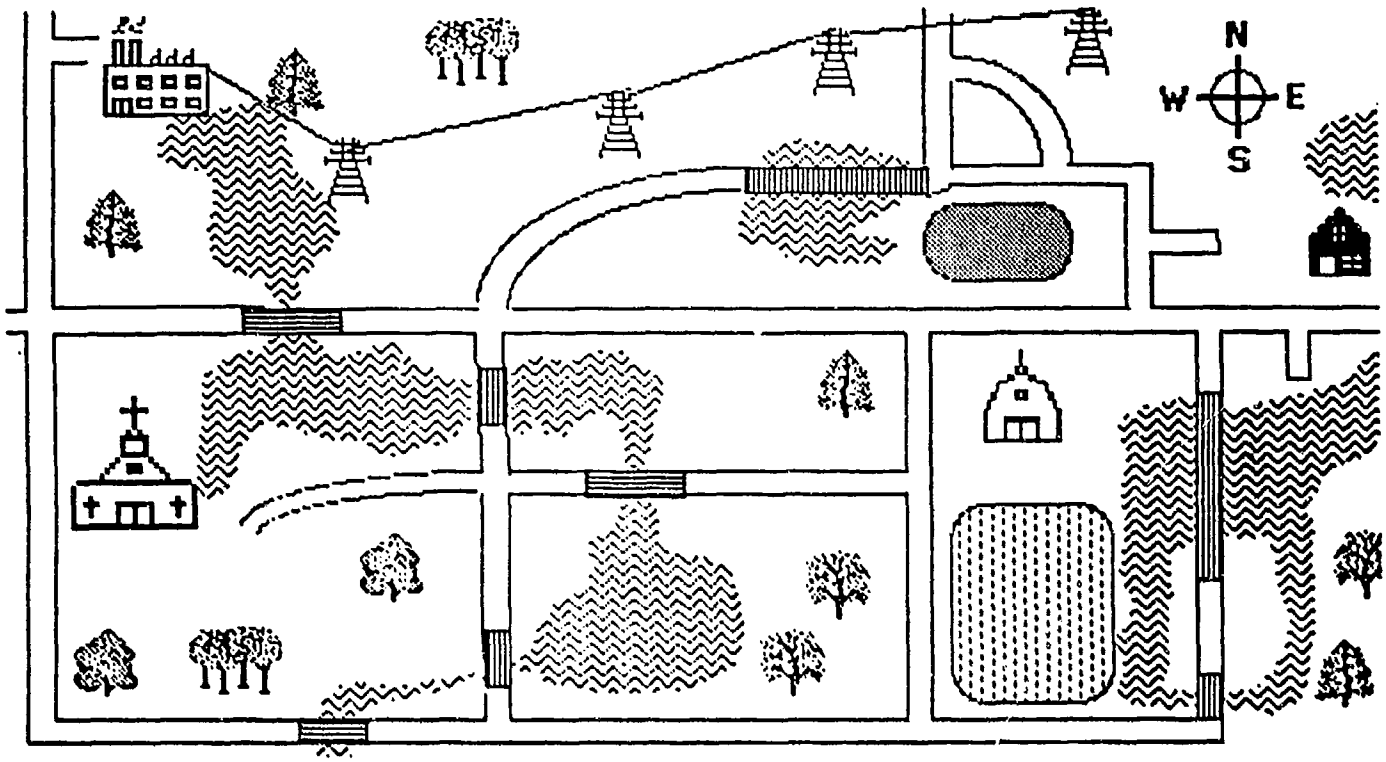


ENGLISH SCRIPT

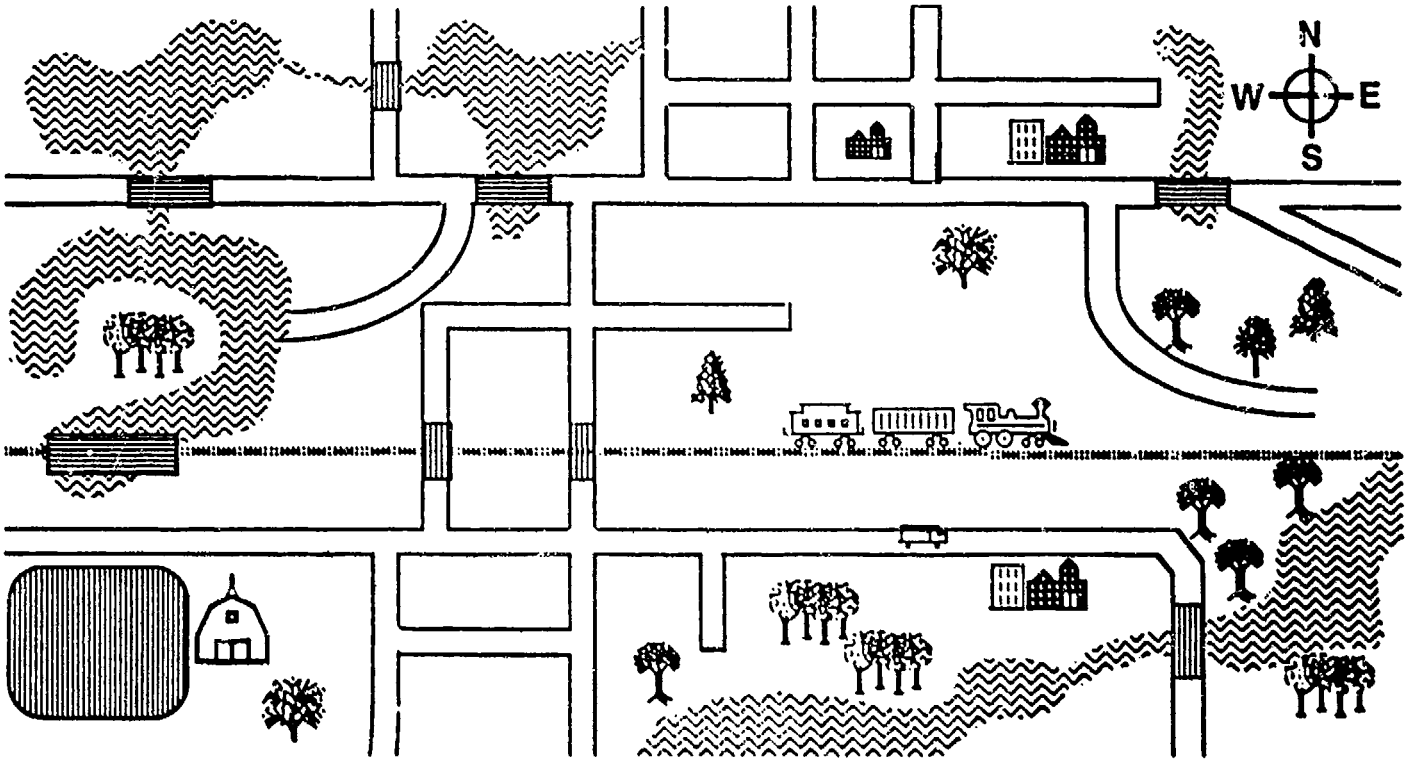
Best Available Copy

APPENDIX C

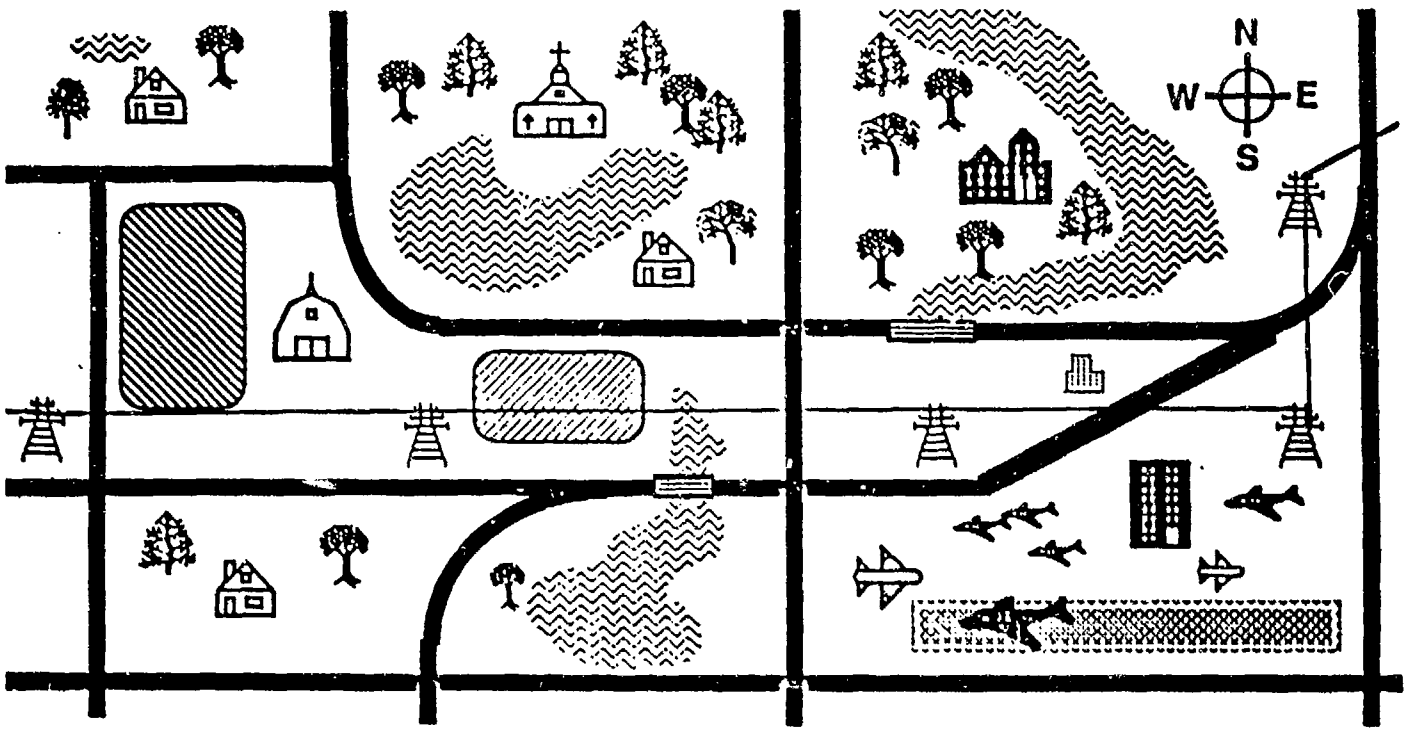
ELECTRONIC MAP BACKGROUNDS



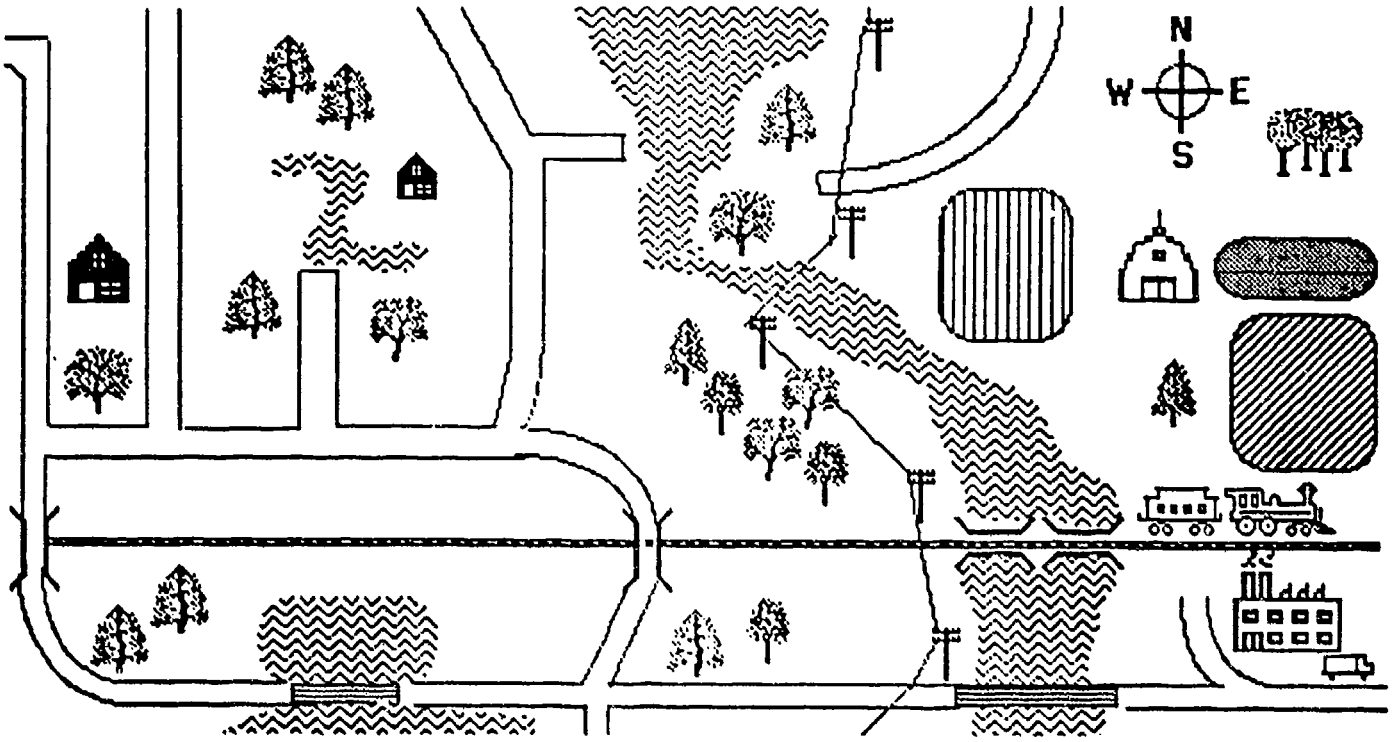
MAP 1



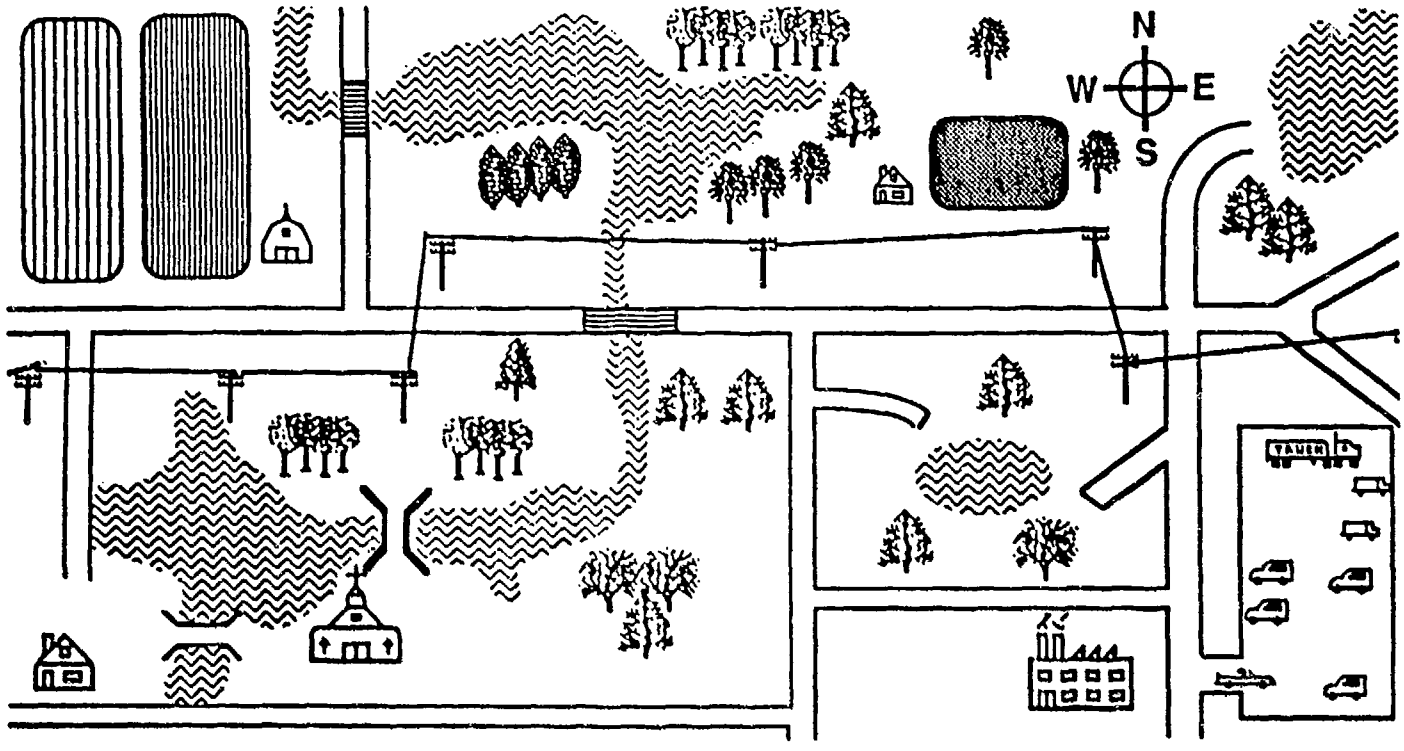
MAP 2



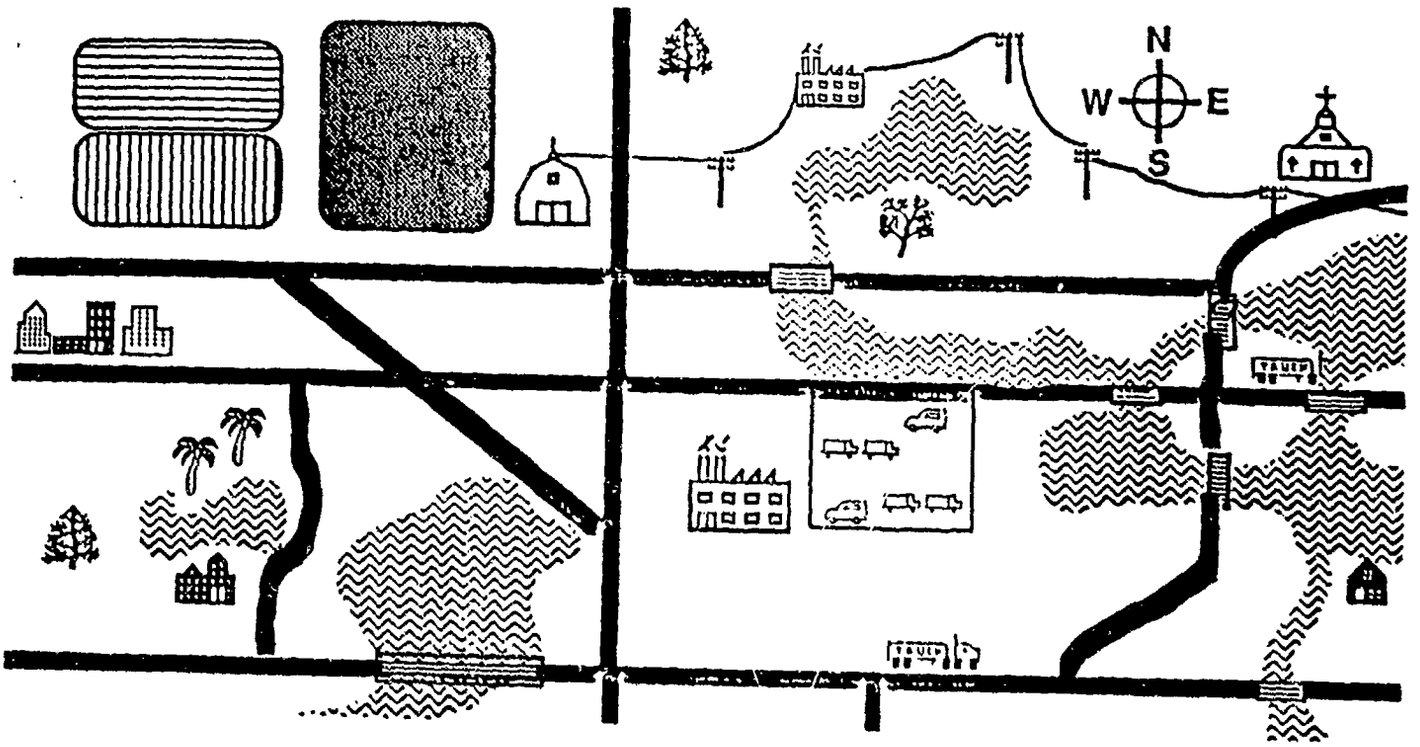
MAP 3



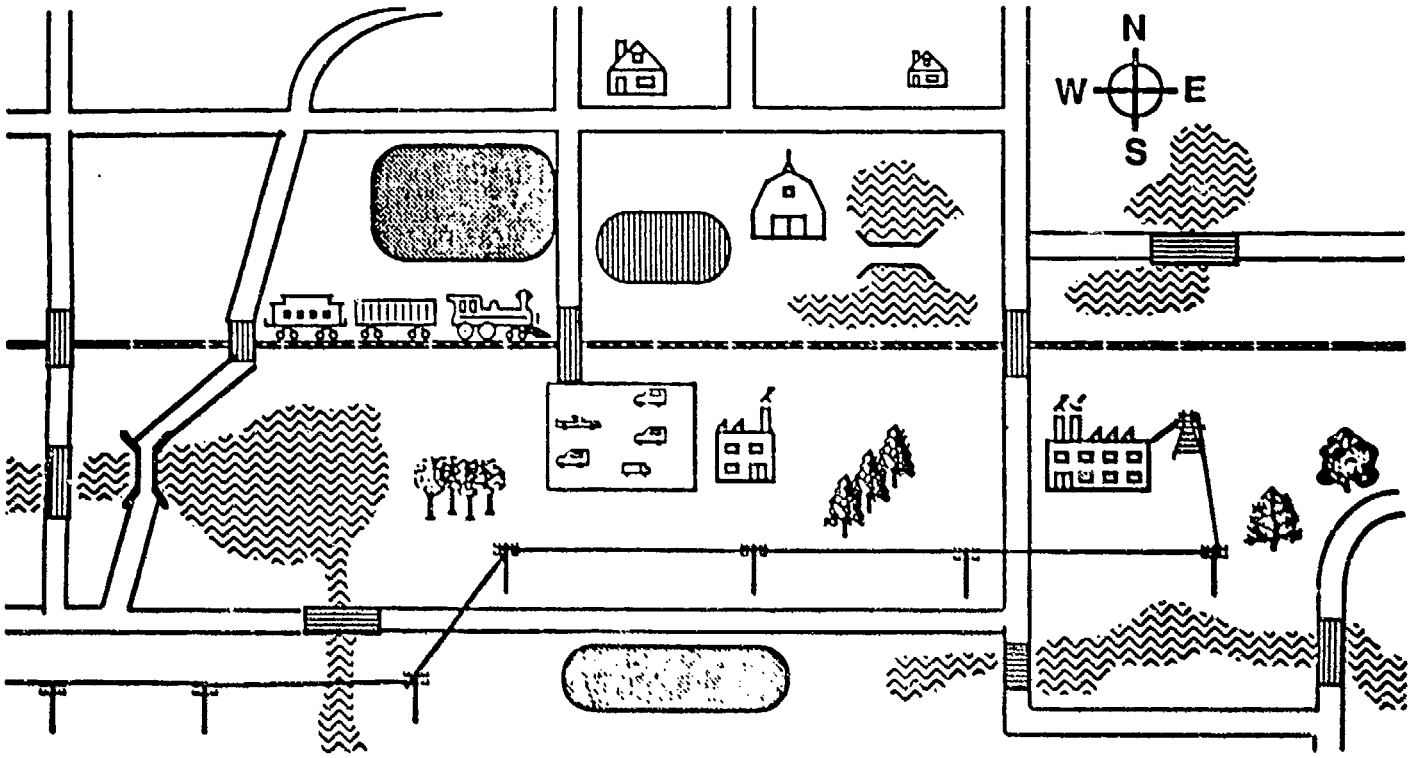
MAP 4



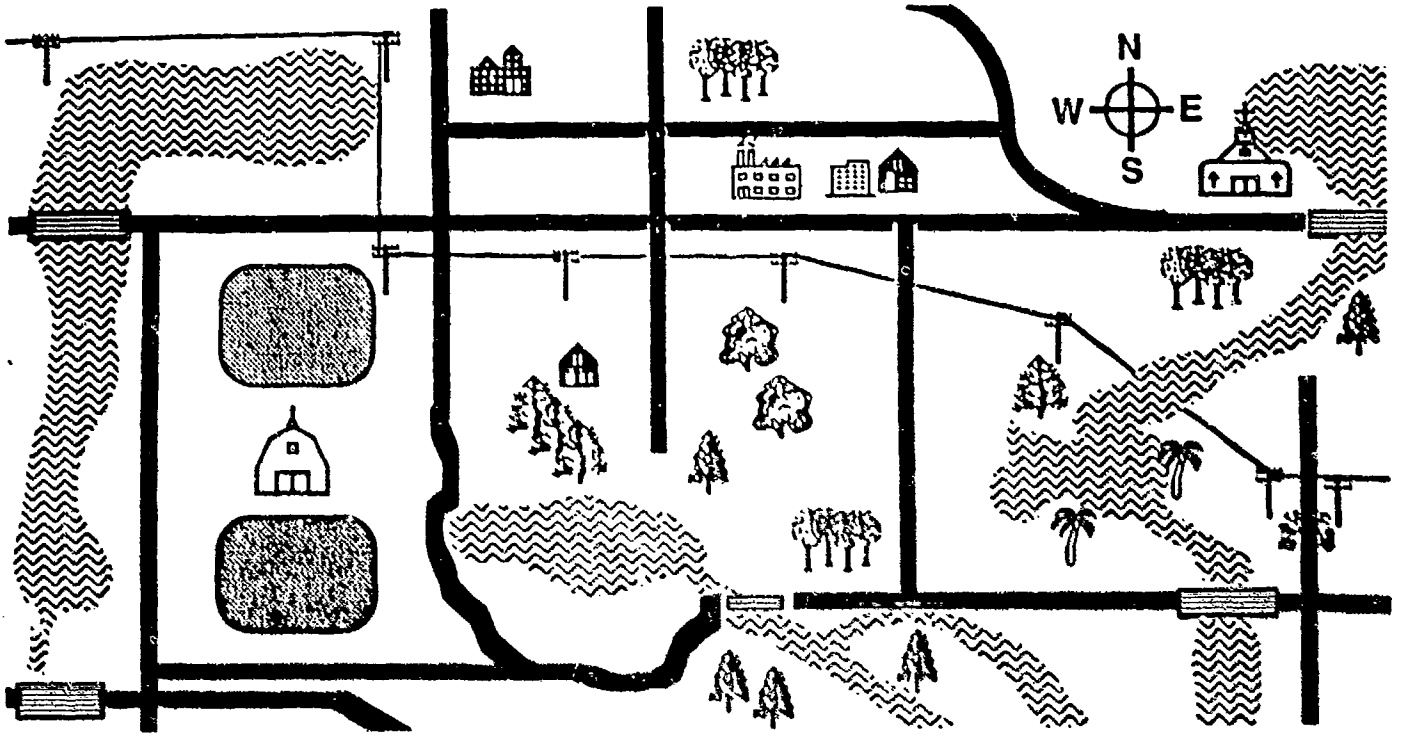
MAP 5



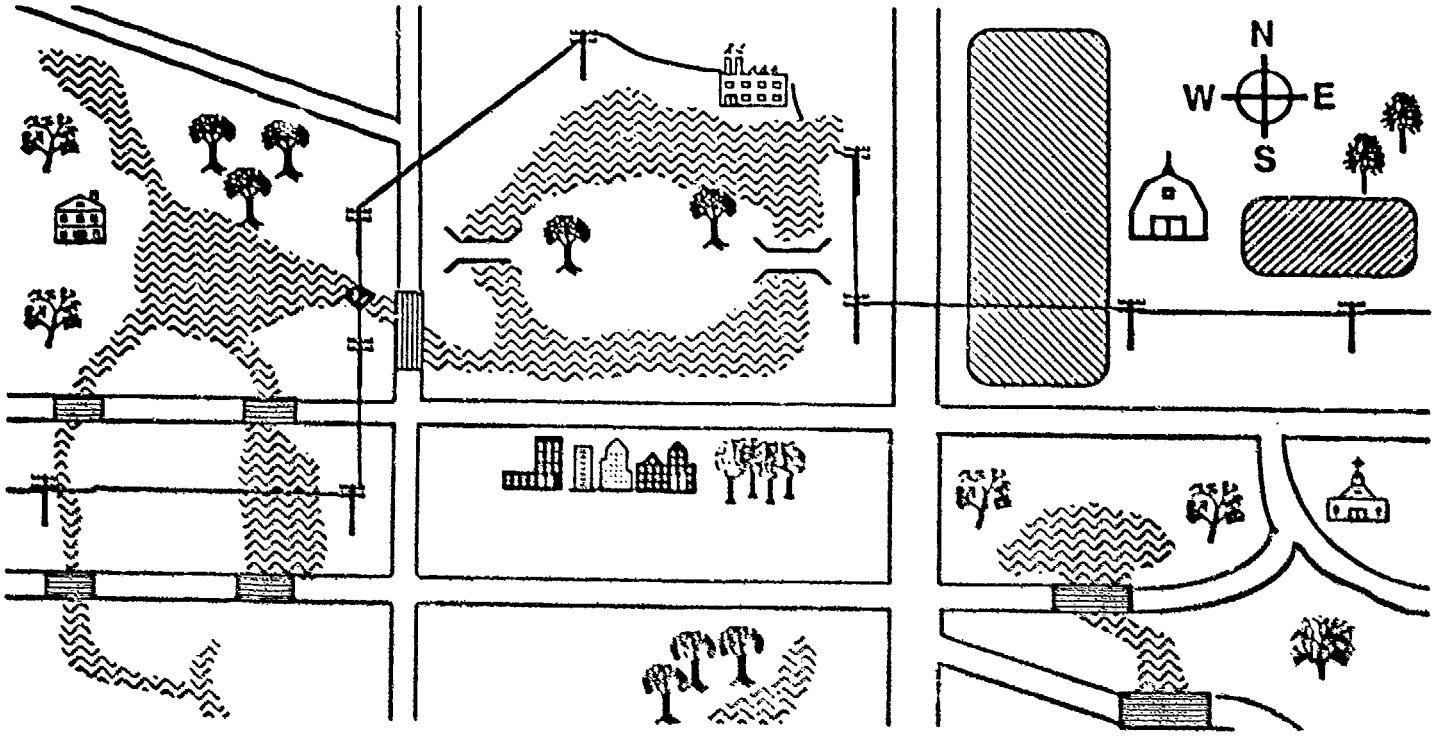
MAP 6



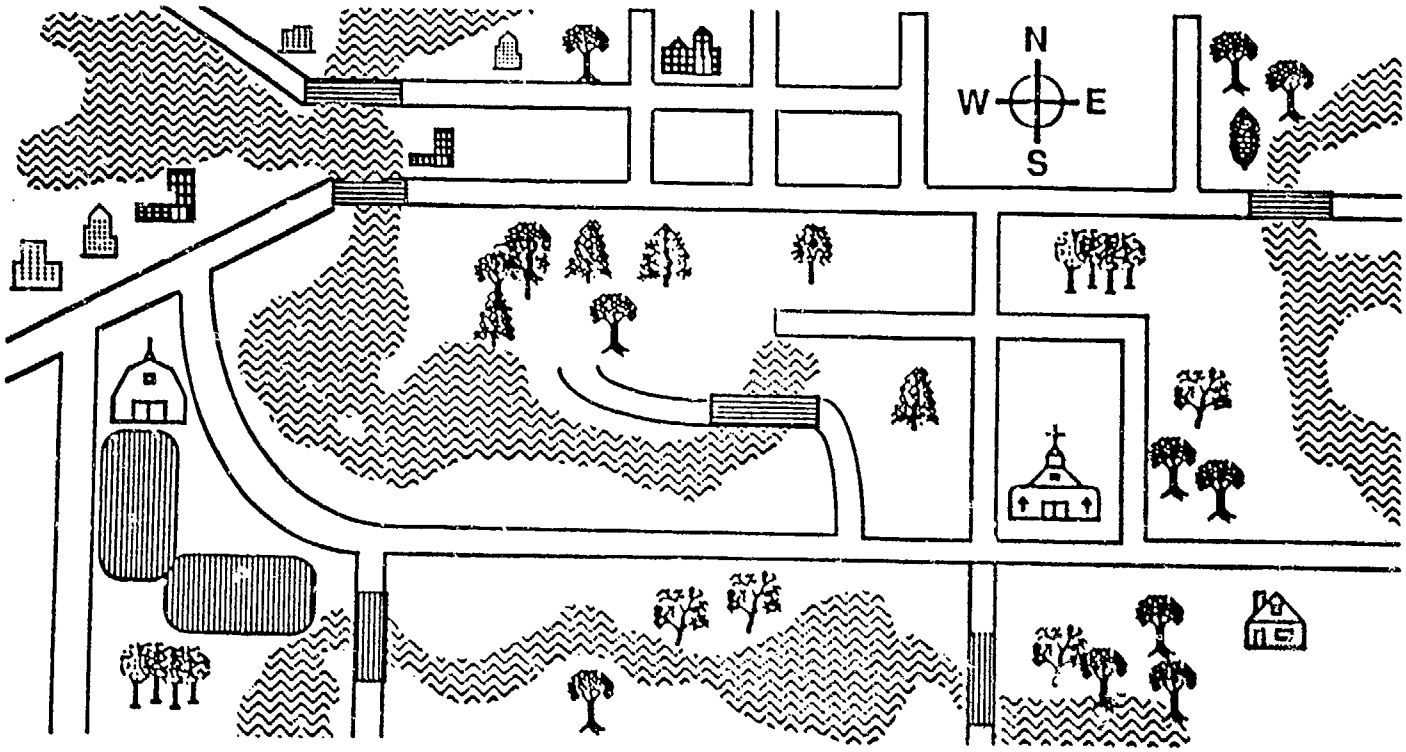
MAP 7



MAP 8



MAP 9



MAP 10

APPENDIX D

TASK PROMPTS AND SCRIPTS



For each trial detailed herein is a three digit code. The first digit, which ranges between 1 and 10, indicates the number of the map which serves as the background (see Appendix C). The second digit, which varies from 1 to 6, indicates the symbol (see Appendix A) which is to be the target on that trial, and the third digit, which also ranges between 1 and 10, specifies the language conditions. These are:

1. Good English script
2. Degraded English script
3. English script for native English speaking subjects (same text as used in 1, above)
4. Spanish script for native Spanish speaking subjects
5. Korean script for native Korean speaking subjects
6. English prompt
7. English prompt for native English speaking subjects (same words as used in 6, above)
8. Spanish prompt for native Spanish speaking subjects
9. Korean prompt for native Korean speaking subjects
10. target symbol iconic prompt

comment: Start at front of church
trial: 1 1 1 Head south and turn east at the road. At the first street turn left. *At the second intersection head east. On the north side of the street, *find the INFANTRY.
trial: 1 1 2 Head south and turn east at the road. Turn left at the first street *and head east at the second intersection. Find the INFANTRY on the *north side of the street.
trial: 1 1 3 Head south and turn east at the road. At the first street turn left. *At the second intersection head east. On the north side of the street, *find the INFANTRY.
trial: 1 1 4 Dirijase al sur y en el camino cruce hacia el este. En la primera calle *cruce la izquierda. En la segunda intersección dirijase al este. *En el lado norte de la calle, encuentre la INFANTERIA.
trial: 1 1 5 남쪽으로 가다 첫번째 만나는 길에서 동쪽으로 가라. *첫번째 교차로에서 좌회전하여 두번째 만나는 사거리에서 *동쪽으로 가다가 이 길 북쪽에있는 보병부대를 찾아라.
trial: 1 1 6 INFANTRY
trial: 1 1 7 INFANTRY
trial: 1 1 8 INFANTERIA
trial: 1 1 9 보병부대
trial: 1 1 10 icon
comment: Start at the front of the barn
trial: 1 2 1 Follow the road west to the crossroads. *Head north and turn at the third street. *Locate the TANK at the end of the road.
trial: 1 2 2 Follow the road west to the crossroads. *Head north and turn at the third street. *At the end of the road locate the TANK.
trial: 1 2 3 Follow the road west to the crossroads. *Head north and turn at the third street. *Locate the TANK at the end of the road.
trial: 1 2 4 Siga el camino hacia el oeste hasta la intersección. *Dirijase al norte y cruce en la tercera calle. *Localice el TANQUE al final del camino.
trial: 1 2 5 서쪽으로 따라가다가 두번째 교차로를 만나면 북쪽으로 가라. *세번째 교차로에서 좌회전하여 길끝에 있는 탱크를 찾아라.
trial: 1 2 6 TANK

trial: 1 2 7 TANK
trial: 1 2 8 TANQUE
trial: 1 2 9 탱크
trial: 1 2 10 icon
comment: Start at front of church
trial: 1 3 1 Head east. At the end of the road turn left, and then left again. *On the north side of the street locate the first GUN.
trial: 1 3 2 Head east and turn left at the end of the road. *Locate the first GUN on the north side of the street after making *the first left turn.
trial: 1 3 3 Head east. At the end of the road turn left, and then left again. *On the north side of the street locate the first GUN.
trial: 1 3 4 Dirijase hacia el este. Al final del camino cruce a la izquierda y luego a la izquierda nuevamente *En el lado norte de la calle localice el primer CAÑON.
trial: 1 3 5 동쪽으로 가서 막다른 골목에서 왼쪽으로 가라. *첫번째 교차로에서 왼쪽으로 가다가 북쪽에있는 대포를 찾아라.
trial: 1 3 6 GUN
trial: 1 3 7 GUN
trial: 1 3 8 CAÑON
trial: 1 3 9 대포
trial: 1 3 10 icon
comment: Start at the road near the powerhouse
trial: 1 4 1 Travel south. At the first crossroad turn left. *On the first cross street head north. *After the bridge locate the ANTITANK GUN.
trial: 1 4 2 Travel south and turn left at the first crossroad. *Head north at the first cross street *and locate the ANTITANK GUN after the bridge.
trial: 1 4 3 Travel south. At the first crossroad turn left. *On the first cross street head north. *After the bridge locate the ANTITANK GUN.
trial: 1 4 4 Viaje hacia el sur. En la primera intersección cruce a la izquierda. *En la primeral transversal dirijase hacia el norte. *Después del puente localice el CAÑON ANTI-TANQUE.
trial: 1 4 5 남쪽으로 가서 첫번째 사거리에서 좌회전하여 첫번째 *사거리에서 북쪽으로 가다가 다리를 건넌후 탱크격파용 *대포를 찾아라.
trial: 1 4 6 ANTITANK GUN

trial: 1 4 7 ANTITANK GUN
trial: 1 4 8 CAÑON ANTI-TANQUE
trial: 1 4 9 탱크격파용 대포
trial: 1 4 10 icon
comment: Start at the front of the barn
trial: 1 5 1 Head south across the field and at the road turn west. *At the second intersection turn right. *After crossing the second bridge locate the HOSPITAL.
trial: 1 5 2 Head south across the field and at the road *and turn west. Turn right at the second intersection *and locate the HOSPITAL after crossing the second bridge.
trial: 1 5 3 Head south across the field and at the road turn west. *At the second intersection turn right. *After crossing the second bridge locate the HOSPITAL.
trial: 1 5 4 Dirijase hacia el sur a través del campo y en el camino cruce al oeste. *En la segunda intersección cruce a la derecha. *Después de cruzar el segundo puente localice el HOSPITAL.
trial: 1 5 5 남쪽으로 밭을 건너서 길을 만나면 서쪽으로 가다가 두번째 *교차로에서 우회전하라. *두개의 다리를 건너가서 병원을 찾아라.
trial: 1 5 6 HOSPITAL
trial: 1 5 7 HOSPITAL
trial: 1 5 8 HOSPITAL
trial: 1 5 9 병원
trial: 1 5 10 icon
comment: Start at the road near the powerhouse
trial: 1 6 1 Head south. At the first intersection turn east. *After crossing the second street, on the south side of the road, *find the PERSONNEL CARRIER.
trial: 1 6 2 Head south and turn east at the first intersection. *After crossing the second street, find the PERSONNEL CARRIER *on the south side of the road.
trial: 1 6 3 Head south. At the first intersection turn east. *After crossing the second street, on the south side of the road, *find the PERSONNEL CARRIER.
trial: 1 6 4 Dirijase hacia al sur. En la primera intersección cruce hacia al este. *Después de cruzar la segunda calle, en el lado sur del camino. *Encuentre el VEHICULO DE TRANSPORTE DE PERSONAL.
trial: 1 6 5 남쪽으로 가서 첫번째 교차로에서 동쪽으로 가라. *두개의 교차로를 지난후 길 남쪽에있는 전령을 찾아라.

trial: 1 6 6 PERSONNEL CARRIER
trial: 1 6 7 PERSONNEL CARRIER
trial: 1 6 8 VEHICULO DE TRANSPORTE DE PERSONAL
trial: 1 6 9 전령
trial: 1 6 10 icon
comment: Start at the east side of the barn
trial: 2 1 1 Go east. At the second street turn left and after the railroad tracks turn right. *Locate the INFANTRY.
trial: 2 1 2 Go east. Turn left at the second street and turn right *after the railroad tracks. Locate the INFANTRY.
trial: 2 1 3 Go east. At the second street turn left and after the railroad tracks turn right. *Locate the INFANTRY.
trial: 2 1 4 Vaya al este. En la segunda calle cruce a la izquierda y después de los rieles del ferrocarril cruce a la derecha. *Localice la INFANTERIA.
trial: 2 1 5 동쪽으로 가서 두번째 교차로에서 좌회전 하라. *그리고 철도를 건너 우회전하여 보병부대를 찾아라.
trial: 2 1 6 INFANTRY
trial: 2 1 7 INFANTRY
trial: 2 1 8 INFANTERIA
trial: 2 1 9 보병부대
trial: 2 1 10 icon
comment: Start at the east side of the barn
trial: 2 2 1 Go east to the end of the road and then north to the end of the road. *Turn west. After the first southbound street locate the TANK.
trial: 2 2 2 Go east to the end of the road and then north *to the end of the road. Turn west and locate the TANK *after the first southbound street.
trial: 2 2 3 Go east to the end of the road and then north to the end of the road. *Turn west. After the first southbound street locate the TANK.
trial: 2 2 4 Vaya al este hasta el final del camino y entonces dirijase hacia el norte hasta el final del camino. *Cruce hacia el oeste. Después de la primera calle que se dirije hacia el sur localice el TANQUE
trial: 2 2 5 동쪽으로 이길의 끝까지 가서 북쪽으로 가라. *그길이 끝나는 곳에서 좌회전하여 첫번째 *만나는 남쪽으로 난 길을 지난후 탱크를 찾아라.

trial: 2 2 6 TANK
trial: 2 2 7 TANK
trial: 2 2 8 TANQUE
trial: 2 2 9 탱크
trial: 2 2 10 icon
comment: Start at the road in the NE corner
trial: 2 3 1 Go west and cross two bridges. *On the south side of the street find the GUN.
trial: 2 3 2 Go west. Find the GUN on the south side of the street *after crossing two bridges.
trial: 2 3 3 Go west and cross two bridges. *On the south side of the street find the GUN.
trial: 2 3 4 Vaya hacia el oeste y cruce dos puentes. *En el lado sur de la calle encuentre el CAÑON.
trial: 2 3 5 서쪽으로 가다가 다리를 두개 건넌후 길 남쪽에 있는 *대포를 찾아라.
trial: 2 3 6 GUN
trial: 2 3 7 GUN
trial: 2 3 8 CAÑON
trial: 2 3 9 대포
trial: 2 3 10 icon
comment: Start at the road in the NE corner
trial: 2 4 1 Head west. Turn onto the third northbound street *and then turn east. *On the north side of the street locate the first ANTITANK GUN.
trial: 2 4 2 Head west. Turn onto the third northbound street *and then turn east. Locate the first ANTITANK GUN *on the north side of the street.
trial: 2 4 3 Head west. Turn onto the third northbound street *and then turn east. *On the north side of the street locate the first ANTITANK GUN.
trial: 2 4 4 Dirijase hacia el oeste. Cruce en la tercera calle que se dirije hacia el norte *y entonces cruce hacia el este. *En lado norte de la calle localice el primer CAÑON ANTI-TANQUE
trial: 2 4 5 서쪽으로 가다가 북쪽으로 향한 세번째 교차로에서 북쪽으로 가라. *첫번째 교차로에서 동쪽으로 향한후 도로 북쪽에 있는 *탱크격파용 대포를 찾아라.
trial: 2 4 6 ANTITANK GUN
trial: 2 4 7 ANTITANK GUN
trial: 2 4 8 CAÑON ANTI-TANQUE
trial: 2 4 9 탱크격파용 대포

trial: 2 4 10 icon
comment: Start at the road in the NE corner
trial: 2 5 1 Travel west and just before the second bridge turn south. *At the third street turn right. *On the north side of the road locate the HOSPITAL.
trial: 2 5 2 Travel west and turn south just before the second bridge. *Turn right at the third street. Locate the HOSPITAL on the north side of the road.
trial: 2 5 3 Travel west and just before the second bridge turn south. *At the third street turn right. *On the north side of the road locate the HOSPITAL.
trial: 2 5 4 Viaje hacia el este y justo antes del segundo puente cruce al sur. *En la tercera calle cruce a la derecha. *En la lado norte del camino localice el HOSPITAL.
trial: 2 5 5 서쪽으로 가다가 두번째 다리전에 남쪽으로 가라. *세번째 교차로에서 우회전하여 북쪽에 있는 병원을 찾아라.
trial: 2 5 6 HOSPITAL
trial: 2 5 7 HOSPITAL
trial: 2 5 8 HOSPITAL
trial: 2 5 9 병원
trial: 2 5 10 icon
comment: Start at the road in the NE corner
trial: 2 6 1 Head west and turn onto the third southbound street. *On the north side of the railroad tracks *locate the PERSONNEL CARRIER.
trial: 2 6 2 Head west and turn onto the third southbound street. *Locate the PERSONNEL CARRIER on the north side *of the railroad tracks.
trial: 2 6 3 Head west and turn onto the third southbound street. *On the north side of the railroad tracks *locate the PERSONNEL CARRIER.
trial: 2 6 4 Dirijase hacia el oeste y cruce en la tercera calle que se drije hacia al sur. *En el lado norte de los rieles del ferrocarril *localice el VEHICULO DE TRANSPORTE DE PERSONAL.
trial: 2 6 5 서쪽으로 가다가 세번째 교차로에서 남쪽으로 가라. *철로 북쪽에 있는 전령을 찾아라.
trial: 2 6 6 PERSONNEL CARRIER
trial: 2 6 7 PERSONNEL CARRIER
trial: 2 6 8 VEHICULO DE TRANSPORTE DE PERSONAL
trial: 2 6 9 전령
trial: 2 6 10 icon

comment: Start at the road at the southwest corner of the airport.
trial: 3 1 1 Travel north and at the second intersection turn west. *Follow this road as it curves around the lake *and find the INFANTRY.
trial: 3 1 2 Travel north and turn west at the second intersection. *Find the INFANTRY on the road *which curves around the lake.
trial: 3 1 3 Travel north and at the second intersection turn west. *Follow this road as it curves around the lake *and find the INFANTRY.
trial: 3 1 4 Viaje hacia el norte y en la segunda intersección cruce hacia el oeste. *Siga éste camino alrededor del lago *y encuentre la INFANTERIA.
trial: 3 1 5 북쪽으로 가서 두번째 교차로에서 좌회전 하라. *그길로 호수를 끼고 돌아가다가 보병부대를 찾아라.
trial: 3 1 6 INFANTRY
trial: 3 1 7 INFANTRY
trial: 3 1 8 INFANTERIA
trial: 3 1 9 보병부대
trial: 3 1 10 icon
comment: Start at the road in the NE corner
trial: 3 2 1 Travel south and bear right. *After passing the power lines take the south fork. *Cross the bridge and find the TANK.
trial: 3 2 2 Travel south and bear right. Take the south fork *after passing the power lines and find the TANK *which is located beyond the bridge.
trial: 3 2 3 Travel south and bear right. *After passing the power lines take the south fork. *Cross the bridge and find the TANK.
trial: 3 2 4 Viaje hacia el sur y permanezca en el lado derecho. Tome la bifurcación sur. *Cruce el puente y encuentre el TANQUE.
trial: 3 2 5 남쪽으로 따라가다가 갈림길에서 우측길로 가라. *송전선을 지난후 갈림길에서 남쪽길로 쭉 따라가다 다리를 건넌후 *탱크를 찾아라.
trial: 3 2 6 TANK
trial: 3 2 7 TANK
trial: 3 2 8 TANQUE
trial: 3 2 9 탱크
trial: 3 2 10 icon

comment: Start at the road in the NE corner
trial: 3 3 1 Go south. Take the right fork and then bear left. *At the crossroads turn south and locate the first GUN.
trial: 3 3 2 Go south. Take the right fork and then bear left. *Locate the first GUN after turning south at the crossroads.
trial: 3 3 3 Go south. Take the right fork and then bear left. *At the crossroads turn south and locate the first GUN.
trial: 3 3 4 Vaya hacia el sur. Tome la bifurcación derecha y permanezca en el lado izquierdo. *En la intersección cruce hacia el sur y localice el primer CAÑON.
trial: 3 3 5 남쪽으로 따라가다가 갈림길에서 우측길로 가면서 또 갈림길이 •나오면 좌측으로 가라. ••사거리에서 남쪽으로 가다가 첫번째 나타나는 대포를 찾아라.
trial: 3 3 6 GUN
trial: 3 3 7 GUN
trial: 3 3 8 CAÑON
trial: 3 3 9 대포
trial: 3 3 10 icon
comment: Start at the road in the NE corner
trial: 3 4 1 Head south and take left fork. After passing the airport turn right *and find the first ANTITANK GUN.
trial: 3 4 2 Head south. Take the left fork and turn right *after passing the airport. Find the first ANTITANK GUN.
trial: 3 4 3 Head south and take left fork. After passing the airport turn right *and find the first ANTITANK GUN.
trial: 3 4 4 Dirijase hacia el sur y tome la bifurcación izquierda. Después de pasar el aereopuerto cruce a la derecha *y encuentre el primer CAÑON ANTI-TANQUE.
trial: 3 4 5 남쪽으로 가다가 갈림길을 만나도 똑바로 가서 비행장을 ••지난후 우회전하여 가면서 첫번째 나타나는 •탱크격파용 •대포를 찾아라.
trial: 3 4 6 ANTITANK GUN
trial: 3 4 7 ANTITANK GUN
trial: 3 4 8 CAÑON ANTI-TANQUE
trial: 3 4 9 •탱크격파용 대포
trial: 3 4 10 icon

comment: Start at the road at the southwest corner of the airport.
trial: 3 5 1 Head north and at the first street turn left. *At the second road head north. *On the west side of the street locate the HOSPITAL.
trial: 3 5 2 Head north and at the first street turn left. *Locate the HOSPITAL on the west side of the second street *as you head north.
trial: 3 5 3 Head north and at the first street turn left. *At the second road head north. *On the west side of the street locate the HOSPITAL.
trial: 3 5 4 Dirijase hacia el norte y en la primera calle cruce a la izquierda. *En el segundo camino dirijase hacia el norte. *En el lado oeste de la calle localice el HOSPITAL.
trial: 3 5 5 북쪽으로 가다가 첫번째 교차로에서 좌회전하라. *두번째 사거리에서 북쪽으로 가다가 길 서쪽에 *있는 병원을 찾아라.
trial: 3 5 6 HOSPITAL
trial: 3 5 7 HOSPITAL
trial: 3 5 8 HOSPITAL
trial: 3 5 9 병원
trial: 3 5 10 icon
comment: Start at the road at the southwest corner of the airport
trial: 3 6 1 Go north. Before reaching the power lines turn left *and then bear right. *After the first intersection locate the PERSONNEL CARRIER.
trial: 3 6 2 Go north. Turn left and then bear right *before reaching the power lines. *Locate the PERSONNEL CARRIER after the first intersection.
trial: 3 6 3 Go north. Before reaching the power lines turn left *and then bear right. *After the first intersection locate the PERSONNEL CARRIER.
trial: 3 6 4 Vaya hacia el norte. Antes de llegar a los cables eléctricos cruce a la izquierda *y entonces permanezca en la derecha. *Después de la primera intersección localice el VEHICULO DE TRANSPORTE DE PERSONAL.
trial: 3 6 5 북쪽으로 따라가다가 송전선을 만나기 전에 좌회전하여 •그 길을 따라가다가 갈림길을 만나도 똑바로 가서 첫번째 •교차로에있는 전령을 찾아라.
trial: 3 6 6 PERSONNEL CARRIER
trial: 3 6 7 PERSONNEL CARRIER

trial: 3 6 8 VEHICULO DE TRANSPORTE DE PERSONAL
trial: 3 6 9 전령
trial: 3 6 10 icon
comment: Start at road in the south-center of the map
trial: 4 1 1 Travel north and at the first street turn right. *Leave the road and head east through the trees. *At the west edge of the river locate the INFANTRY.
trial: 4 1 2 Travel north and turn right at the first street. *Leave the road walk through the trees from the west. *Locate the INFANTRY at the west edge of the river.
trial: 4 1 3 Travel north and at the first street turn right. *Leave the road and head east through the trees. *At the west edge of the river locate the INFANTRY.
trial: 4 1 4 Viaje hacia el norte y en la primera calle cruce hacia la derecha. *Salgase del camino y vaya hacia el este a través de los árboles. *En la orilla oeste del río localice la INFANTERIA.
trial: 4 1 5 북쪽으로 가서 첫번째 교차로에서 길을 벗어나 동쪽으로 가다가 숲을 통과해서 강 서편 기슭에있는 보병부대를 찾아라.
trial: 4 1 6 INFANTRY
trial: 4 1 7 INFANTRY
trial: 4 1 8 INFANTERIA
trial: 4 1 9 보병부대
trial: 4 1 10 icon
comment: Start at road in the south-center of the map
trial: 4 2 1 Head north. At the third intersection turn north. *On the east side of the street find the TANK.
trial: 4 2 2 Head north and then turn north at the third intersection. *Find the TANK on the east side of the street.
trial: 4 2 3 Head north. At the third intersection turn north. *On the east side of the street find the TANK.
trial: 4 2 4 Dirijase hacia el norte. En la tercera intersección cruce al norte. *En el lado este de la calle encuentre el TANQUE.
trial: 4 2 5 북쪽으로 가다가 세번째 교차로를 만나면 그 북쪽으로 가라. *길동쪽에 있는 탱크를 찾아라.
trial: 4 2 6 TANK
trial: 4 2 7 TANK

trial: 4 2 8 TANQUE
trial: 4 2 9 탱크
trial: 4 2 10 icon
comment: Start at the road entering from the west edge
trial: 4 3 1 Follow the road south and at the first road turn left. *At the railroad tracks head west. *On the left side of the tracks locate the first GUN.
trial: 4 3 2 Follow the road south and turn left at the first road. *Head west at the railroad tracks and locate the first GUN *on the left side of the tracks.
trial: 4 3 3 Follow the road south and at the first road turn left. *At the railroad tracks head west. *On the left side of the tracks locate the first GUN.
trial: 4 3 4 Siga el camino hacia sur y en el primer camino cruce a la izquierda. *Al llegar a los rieles de ferrocarril dirijase hacia el oeste. *En el lado izquierdo de los rieles del ferrocarril localice el primer CAÑON.
trial: 4 3 5 남쪽으로 가서 첫번째 교차로에서 좌회전하라 *철교를 만나면 철도를 따라 서쪽으로 가다가 좌측편에있는 *대포를 찾아라
trial: 4 3 6 GUN
trial: 4 3 7 GUN
trial: 4 3 8 CAÑON
trial: 4 3 9 대포
trial: 4 3 10 icon
comment: Start at road in the south-center of the map
trial. 4 4 1 Go north and then follow the railroad tracks eastward. *After the double bridge head left and then turn west at the barn. *At the north edge of the field indicate the ANTITANK GUN.
trial: 4 4 2 Go north and then follow the railroad tracks eastward. *Head left after the double bridge and then at the barn turn west. *Indicate the ANTITANK GUN at the north edge of the field.
trial: 4 4 3 Go north and then follow the railroad tracks eastward. *After the double bridge head left and then turn west at the barn. *At the north edge of the field indicate the ANTITANK GUN.
trial: 4 4 4 Vaya hacia el norte y siga los rieles del ferrocarril en dirección este. *Después del puente-doble dirijase a la izquierda y luego cruce al oeste en en establo. *En el lado norte del campo señale el CAÑON ANTI-TANQUE.

trial: 4 4 5 북쪽으로 가다가 철교를 만나면 철로를 따라 동쪽으로 가라. *쌍철교를 지나자마자 왼쪽으로 가다가 굿간을 만나면 그곳에서 왼쪽으로 가서 밭 북쪽에 있는 탱크격파용 대포를 찾아라.
trial: 4 4 6 ANTITANK GUN
trial: 4 4 7 ANTITANK GUN
trial: 4 4 8 CAÑON ANTI-TANQUE
trial: 4 4 9 탱크격파용 대포
trial: 4 4 10 icon
comment: Start at the road entering from the west edge in the NW corner
trial: 4 5 1 Travel south and at the first intersection turn left. *After the second street find the first HOSPITAL.
trial: 4 5 2 Travel south. At the first intersection turn left *and find the first HOSPITAL after the second street.
trial: 4 5 3 Travel south and at the first intersection turn left. *After the second street find the first HOSPITAL.
trial: 4 5 4 Viaje hacia el sur y la primera intersección cruce a la izquierda. *Después de la segunda calle encuentre el primer HOSPITAL.
trial: 4 5 5 남쪽으로 가서 첫번째 교차로에서 좌회전하라. *두번째 교차로를 지나 첫번째 나타나는 병원을 찾아라.
trial: 4 5 6 HOSPITAL
trial: 4 5 7 HOSPITAL
trial: 4 5 8 HOSPITAL
trial: 4 5 9 병원
trial: 4 5 10 icon
comment: Start at the road entering from the west edge in the NW corner
trial: 4 6 1 Head south and at the second intersection turn left. *After crossing the railroad tracks find the first PERSONNEL CARRIER.
trial: 4 6 2 Head south and turn left at the second intersection. *Find the first PERSONNEL CARRIER after crossing the railroad tracks.
trial: 4 6 3 Head south and at the second intersection turn left. *After crossing the railroad tracks find the first PERSONNEL CARRIER.
trial: 4 6 4 Dirijase al sur y en la segunda intersección cruce a la izquierda. *Después de cruzar los rieles del

ferrocarril encuentre el primer VEHICULO DE TRANSPORTE DE PERSONAL.

trial: 4 6 5 남쪽으로 가다가 두번째 교차로에서 좌회전하여 철로를 지난후 첫번째 전령을 찾아라.
trial: 4 6 6 PERSONNEL CARRIER
trial: 4 6 7 PERSONNEL CARRIER
trial: 4 6 8 VEHICULO DE TRANSPORTE DE PERSONAL
trial: 4 6 9 전령
trial: 4 6 10 icon
comment: Start at the northern most road entering from the west edge
trial: 5 1 1 Travel east to the third intersection and turn south. *Turn again at the dead end street. *On the north side of the road locate the INFANTRY.
trial: 5 1 2 Travel east and turn south at the third intersection. *Turn again at the dead end street. *Locate the INFANTRY on the north side of the road.
trial: 5 1 3 Travel east to the third intersection and turn south. *Turn again at the dead end street. *On the north side of the road locate the INFANTRY.
trial: 5 1 4 Viaje hacia el este hasta la tercera intersección y cruce hacia el sur. *Cruce nuevamente al final de la calle ciega. *En el lado norte del camino localice la INFANTERIA.
trial: 5 1 5 동쪽으로 가서 세번째 교차로에서 남쪽으로 가다가 다시 *교차로에서 좌회전하여 막다른길 북쪽에있는 보병부대를 찾아라.
trial: 5 1 6 INFANTRY
trial: 5 1 7 INFANTRY
trial: 5 1 8 INFANTERIA
trial: 5 1 9 보병부대
trial: 5 1 10 icon
comment: Start at the northern most road entering from the east edge
trial: 5 2 1 Head west and cross the power lines twice. *Turn at the first northbound street. *After the bridge locate the TANK.
trial: 5 2 2 Head west. Turn at the first northbound street *after crossing the power lines twice. *Locate the TANK after the bridge.
trial: 5 2 3 Head west and cross the power lines twice. *Turn at the first northbound street. *After the bridge locate the TANK.

trial: 5 2 4 Dirijase hacia el oeste y cruce los cables eléctricos dos veces. *Cruce en la primera calle que va hacia el norte. *Después del puente localice el TANQUE.
trial: 5 2 5 서쪽으로 따라가다가 송전선을 두번 지난후 북쪽으로 뻗어있는 *길을 따라가라. 다리를 지나서 탱크를 찾아라.
trial: 5 2 6 TANK
trial: 5 2 7 TANK
trial: 5 2 8 TANQUE
trial: 5 2 9 탱크
trial: 5 2 10 icon
comment: Start at the northern most road entering from the east edge
trial: 5 3 1 Travel west. Cross the bridge and turn at the first street. *On the west side of the road select the GUN.
trial: 5 3 2 Travel west. Turn east at the first street *after crossing the bridge. *Select the GUN on the west side of the road.
trial: 5 3 3 Travel west. Cross the bridge and turn at the first street. *On the west side of the road select the GUN.
trial: 5 3 4 Viaje hacia el oeste. Atraviese el puente y cruce en la primera calle. *En el lado oeste del camino seleccione el CAÑON.
trial: 5 3 5 서쪽으로 가다가 다리를 지나 첫번째 교차로에서 북쪽으로 가다가 좌측편에 있는 대포를 찾아라.
trial: 5 3 6 GUN
trial: 5 3 7 GUN
trial: 5 3 8 CAÑON
trial: 5 3 9 대포
trial: 5 3 10 icon
comment: Start at the northern most road entering from the west edge
trial: 5 4 1 Go east and continue past the power lines. *At the first street head south and then on the first through-street turn again. *Find the ANTITANK GUN.
trial: 5 4 2 Go east. Head south at the first street past the power lines and then *find the ANTITANK GUN after turning on the first through-street.
trial: 5 4 3 Go east and continue past the power lines. *At the first street head south and then on the first through-street turn again. *Find the ANTITANK GUN.

trial: 5 4 4 Vaya hacia al este y continue hasta pasar los cables eléctricos. *En la primera calle dirijase hacia el sur y entonces en la primera transversal cruce otra vez. *Encuentre el CAÑON ANTI-TANQUE.
trial: 5 4 5 동쪽으로 가서 송전선을 지난후 첫번째 교차로에서 *남쪽으로 가다 첫번째 만나는 다른쪽으로 트여있는 길로 접어든후 *탱크격파용 대포를 찾아라.
trial: 5 4 6 ANTITANK GUN
trial: 5 4 7 ANTITANK GUN
trial: 5 4 8 CAÑON ANTI-TANQUE
trial: 5 4 9 탱크격파용 대포
trial: 5 4 10 icon
comment: Start at the northern most road entering from the east edge
trial: 5 5 1 Go west and pass the first set of power lines. *On the first street turn south and then turn again onto the westbound road. *At the west side of the building indicate the HOSPITAL.
trial: 5 5 2 Go west and pass the first set of power lines. *Turn south on the first street and at the westbound road turn again. *Indicate the HOSPITAL at the west side of the building.
trial: 5 5 3 Go west and pass the first set of power lines. *On the first street turn south and then turn again onto the westbound road. *At the west side of the building indicate the HOSPITAL.
trial: 5 5 4 Vaya hacia el oeste y pase el primer conjunto de cables eléctricos. *En la primera calle cruce hacia el sur y entonces cruce otra vez en el camino que va a al oeste. *Al lado oeste del edificio señale el HOSPITAL.
trial: 5 5 5 서쪽으로 가서 첫번째 송전선을 지난후 첫번째 교차로에서 *남쪽으로 가다 서쪽으로 뻗어있는 길로 접어든 후 교회의 서쪽에 있는 병원을 찾아라.
trial: 5 5 6 HOSPITAL
trial: 5 5 7 HOSPITAL
trial: 5 5 8 HOSPITAL
trial: 5 5 9 병원
trial: 5 5 10 icon
comment: Start at the northern most road entering from the west edge

trial: 5 6 1 Head east. Turn at the first road *and then cross the footbridge going east. *At the east edge of the trees find the PERSONNEL CARRIER.
trial: 5 6 2 Head east. Cross the footbridge going east *after turning at the first road. *Find the PERSONNEL CARRIER at the east edge of the trees.
trial: 5 6 3 Head east. Turn at the first road *and then cross the footbridge going east. *At the east edge of the trees find the PERSONNEL CARRIER.
trial: 5 6 4 Dirijase hacia el este. Cruce en el primer camino *y entonces atraviese el puente para peatones que va hacia el este. *Al este de los árboles encuentre el VEHICULO DE TRANSPORTE DE PERSONAL.
trial: 5 6 5 동쪽으로 가서 첫번째 교차로에서 우회전하여 가라. *동쪽으로 향해있는 다리를 건너 숲 동쪽 있는 전령을 찾아라.
trial: 5 6 6 PERSONNEL CARRIER
trial: 5 6 7 PERSONNEL CARRIER
trial: 5 6 8 VEHICULO DE TRANSPORTE DE PERSONAL
trial: 5 6 9 전령
trial: 5 6 10 icon
comment: Start at the northern most road entering from the east edge
trial: 6 1 1 Travel west and at the first bridge turn right. *At the second intersection turn left and on the south side of the street *locate the INFANTRY.
trial: 6 1 2 Travel west and turn right at the first bridge. *Turn left at the second intersection and locate the INFANTRY *on the south side of the street.
trial: 6 1 3 Travel west and at the first bridge turn right. *At the second intersection turn left and on the south side of the street *locate the INFANTRY.
trial: 6 1 4 Viaje hacia el oeste y en el primer puente cruce a la derecha. *El la segunda intersección cruce hacia a la izquierda y en el lado sur de la calle *localice la INFANTERIA.
trial: 6 1 5 서쪽으로 가서 첫번째 다리에서 우회전하라. *두번째 교차로에서 좌회전하여 가다가 도로 남쪽에있는 *보병부대를 찾아라.
trial: 6 1 6 INFANTRY
trial: 6 1 7 INFANTRY
trial: 6 1 8 INFANTERIA
trial: 6 1 9 보병부대

trial: 6 1 10 icon
comment: Start at the road entering from the north edge
trial: 6 2 1 Head south to the second crossroads and turn west. *At the second street head south. *On the west side of the road locate the TANK.
trial: 6 2 2 Head south and turn west at the second crossroads. *Head south at the second street. *Locate the TANK on the west side of the road.
trial: 6 2 3 Head south to the second crossroads and turn west. *At the second street head south. *On the west side of the road locate the TANK.
trial: 6 2 4 Dirijase hacia el sur hasta la segunda intersección y cruce hacia el oeste. *En la segunda calle dirijase hacia el sur. *En el lado oeste del camino localice el TANQUE.
trial: 6 2 5 남쪽으로 따라가다가 두번째 교차로를 만나면 *서쪽으로 가라. 또 두번째 교차로에서 남쪽으로 가서 도로의 *서쪽에 있는 탱크를 찾아라.
trial: 6 2 6 TANK
trial: 6 2 7 TANK
trial: 6 2 8 TANQUE
trial: 6 2 9 탱크
trial: 6 2 10 icon
comment: Start at the road entering from the north edge
trial: 6 3 1 Travel south and at the third crossroad turn east. *Bear right at the fork. *After the bridge indicate the first GUN.
trial: 6 3 2 Travel south and turn east at the third crossroad. *Bear right at the fork. *Indicate the first GUN after the bridge.
trial: 6 3 3 Travel south and at the third crossroad turn east. *Bear right at the fork. *After the bridge indicate the first GUN.
trial: 6 3 4 Viaje hacia el sur y en la tercera intersección cruce hacia el este. *Permanezca en el lado derecho al llegar a la bifurcación. *Después del puente señale el primer CAÑON.
trial: 6 3 5 남쪽으로 가서 세번째 사거리에서 동쪽으로 가라. *두번째 교차로에서 좌회전하여 가다가 다리를 건넌후 *우측편에있는 대포를 찾아라.
trial: 6 3 6 GUN
trial: 6 3 7 GUN
trial: 6 3 8 CAÑON
trial: 6 3 9 대포

trial: 6 3 10 icon
comment: Start at the northern most road entering from the east edge
trial: 6 4 1 Follow the road southwest and continue south through the intersection to the dead end. *Turn right, and right again. At the second intersection turn left *and find the ANTITANK GUN.
trial: 6 4 2 Follow the road southwest and continue south through the intersection to the dead end. *Turn right, and right again. *Find the ANTITANK GUN after turning left at the second intersection.
trial: 6 4 3 Follow the road southwest and continue south through the intersection to the dead end. *Turn right, and right again. At the second intersection turn left *and find the ANTITANK GUN.
trial: 6 4 4 Siga el camino hacia el suroeste hasta llegar al final. *Cruce a la derecha y luego nuevamente a la derecha. En la segunda intersección cruce a la izquierda *y encuentre el CAÑON ANTI-TANQUE.
trial: 6 4 5 서쪽으로 가다가 남쪽으로 막다른 길까지가서 우회전 하라. *두번째 교차로에서 우회전 한후, 또 두번째 교차로에서 좌회전하라. *우측에 있는 탱크격파용 대포를 찾아라.
trial: 6 4 6 ANTITANK GUN
trial: 6 4 7 ANTITANK GUN
trial: 6 4 8 CAÑON ANTI-TANQUE
trial: 6 4 9 탱크격파용 대포
trial: 6 4 10 icon
comment: Start at the road entering from the north edge
trial: 6 5 1 Go south to the second street and turn left. *After crossing the lake turn right. *On the left side of the road find the HOSPITAL.
trial: 6 5 2 Go south and turn left at the second street. *Turn right after crossing the lake *and find the HOSPITAL on the left side of the road.
trial: 6 5 3 Go south to the second street and turn left. *After crossing the lake turn right. *On the left side of the road find the HOSPITAL.
trial: 6 5 4 Vaya hacia el sur hasta la segunda calle y cruce a la izquierda. *Después de pasar el lago cruce a la derecha. *En el lado izquierdo del camino encuentre el HOSPITAL.

trial: 6 5 5 남쪽으로 가다가 두번째 교차로에서 좌회전하라. *호수를 지나자마자 우회전하여 도로의 좌측에 있는 병원을 찾아라.
trial: 6 5 6 HOSPITAL
trial: 6 5 7 HOSPITAL
trial: 6 5 8 HOSPITAL
trial: 6 5 9 병원
trial: 6 5 10 icon
comment: Start at the northern most road entering from the east edge
trial: 6 6 1 Head west and after the first bridge turn right. *At the northwest corner of the factory parking lot *locate the PERSONNEL CARRIER.
trial: 6 6 2 Head west and turn right after the first bridge. *Locate the PERSONNEL CARRIER at the northwest corner *of the factory parking lot.
trial: 6 6 3 Head west and after the first bridge turn right. *At the northwest corner of the factory parking lot *locate the PERSONNEL CARRIER.
trial: 6 6 4 Dirijase hacia el oeste y después del primer puente cruce a la derecha. *En la esquina noroeste del estacionamiento de la fábrica *localice el VEHICULO DE TRANSPORTE DE PERSONAL
trial: 6 6 5 서쪽으로 가서 첫번째 다리를 지난후 우회전 하라. 공장 *주차장 북서쪽 모퉁이에 있는 전령을 찾아라.
trial: 6 6 6 PERSONNEL CARRIER
trial: 6 6 7 PERSONNEL CARRIER
trial: 6 6 8 VEHICULO DE TRANSPORTE DE PERSONAL
trial: 6 6 9 전령
trial: 6 6 10 icon
comment: Start at the northern most road entering from the east edge
trial: 7 1 1 Head west to the dead end and turn left. *Before the second bridge turn right. *On the north side of the road locate the INFANTRY.
trial: 7 1 2 Head west. Turn left at the dead end *and left before the second bridge. *Locate the INFANTRY on the north side of the road.
trial: 7 1 3 Head west to the dead end and turn left. *Before the second bridge turn right. *On the north side of the road locate the INFANTRY.

trial: 7 1 4 Dirijase hacia el oeste hasta el final del camino y cruce a la izquierda. *Antes de llegar al segundo puente cruce a la derecha. *En el lado norte del camino localice la INFANTERIA.

trial: 7 1 5 서쪽으로 가서 막다른데 까지가서 길에서 좌회전하라. *두번째 다리를 지나기 전에 우회전 하라. *길 북쪽에있는 보병부대를 찾아라.

trial: 7 1 6 INFANTRY

trial: 7 1 7 INFANTRY

trial: 7 1 8 INFANTERIA

trial: 7 1 9 보병부대

trial: 7 1 10 icon

comment: Start at the northern most road entering from the east edge

trial: 7 2 1 Travel west to the dead end. Continue west across lake *and past the barn. At the second street turn north. *On west side of road find the TANK.

trial: 7 2 2 Travel west acoss the road and lake. *Turn north at the second street after passing the barn *and find the TANK on west side of road.

trial: 7 2 3 Travel west to the dead end. Continue west across lake *and past the barn. At the second street turn north. *On west side of road find the TANK.

trial: 7 2 4 Viaje hacia el oeste hasta el final del camino. Continúe hacia el oeste a través del lago *y pase el establo. En la segunda calle cruce hacia el norte. *En el lado oeste del camino encuentre el TANQUE.

trial: 7 2 5 서쪽으로 가서 막다른데 까지가서 길을 벗어나 계속 서쪽으로 가라. *호수를 건너고 곳간을 지나서 두번째 길을 만나면 북쪽으로 가라. *길 좌측에 있는 탱크를 찾아라.

trial: 7 2 6 TANK

trial: 7 2 7 TANK

trial: 7 2 8 TANQUE

trial: 7 2 9 탱크

trial: 7 2 10 icon

comment: Start at road entering from the north near the west edge

trial: 7 3 1 Head south and at the first crossroads turn east. *At the second street turn right. *After crossing the bridge locate the GUN.

trial: 7 3 2 Head south. Turn east at the first crossroads *and right at the second street. *Locate the GUN aftercrossing the bridge.
trial: 7 3 3 Head south and at the first crossroads turn east. *At the second street turn right. *After crossing the bridge locate the GUN.
trial: 7 3 4 Dirijase hacia el sur y en la primera intersección cruce hacia el este. *En la segunda calle cruce a la derecha. *Después de pasar el puente localice el CAÑON.
trial: 7 3 5 남쪽으로 가서 첫번째 사거리에서 동쪽으로 가라. *두번째 교차로에서 우회전 한후 다리를 건너자마자 대포를 찾아라.
trial: 7 3 6 GUN
trial: 7 3 7 GUN
trial: 7 3 8 CAÑON
trial: 7 3 9 대포
trial: 7 3 10 icon
comment: Start at road entering from the north near the west edge
trial: 7 4 1 Travel south and cross two bridges. *On the east side of the road locate the ANTITANK GUN.
trial: 7 4 2 Travel south and locate the ANTITANK GUN *on the east side of the road after crossing two bridges.
trial: 7 4 3 Travel south and cross two bridges. *On the east side of the road locate the ANTITANK GUN.
trial: 7 4 4 Viaje hacia el sur y atraviese dos puentes. *En el lado oeste del camino localice el CAÑON ANTI-TANQUE.
trial: 7 4 5 남쪽으로 가서 두번째 다리를 건넌후 길동쪽에 있는 탱크격파용 대포를 찾아라.
trial: 7 4 6 ANTITANK GUN
trial: 7 4 7 ANTITANK GUN
trial: 7 4 8 CAÑON ANTI-TANQUE
trial: 7 4 9 탱크격파용 대포
trial: 7 4 10 icon
comment: Start at road entering from the north near the west edge
trial: 7 5 1 Travel south. At the first street turn left. *After passing the barn find the second HOSPITAL.
trial: 7 5 2 Travel south and turn left at the first street. *Find the second HOSPITAL after passing the barn.

trial: 7 5 3 Travel south. At the first street turn left. *After passing the barn find the second HOSPITAL.
trial: 7 5 4 Viaje hacia el sur. En la primera calle cruce a la izquierda. *Después de pasar el establo encuentre el segundo HOSPITAL.
trial: 7 5 5 남쪽으로 가다가 첫번째 교차로에서 좌회전하라. *헛간을 지난후 두번째 병원을 찾아라.
trial: 7 5 6 HOSPITAL
trial: 7 5 7 HOSPITAL
trial: 7 5 8 HOSPITAL
trial: 7 5 9 병원
trial: 7 5 10 icon
comment: Start at road entering from the north near the west edge
trial: 7 6 1 Go south and at the second intersection turn left. *Pass the bridge and power lines *and find the first PERSONNEL CARRIER.
trial: 7 6 2 Go south. Turn left at the second intersection and find the first PERSONNEL CARRIER *after passing the bridge and power lines.
trial: 7 6 3 Go south and at the second intersection turn left. *Pass the bridge and power lines *and find the first PERSONNEL CARRIER.
trial: 7 6 4 Vaya hacia el sur y en la segunda intersección cruce a la izquierda. *Pase el puente y los cables eléctricos *y encuentre el primer VEHICULO DE TRANSPORTE DE PERSONAL.
trial: 7 6 5 남쪽으로 가서 두번째 교차로에서 좌회전하라. *다리를 건너고 송전선을 지나 첫번째 전령을 찾아라.
trial: 7 6 6 PERSONNEL CARRIER
trial: 7 6 7 PERSONNEL CARRIER
trial: 7 6 8 VEHICULO DE TRANSPORTE DE PERSONAL
trial: 7 6 9 전령
trial: 7 6 10 icon
comment: Start at the southern most road entering from the east edge
trial: 8 1 1 Travel west and at the third road turn right. *North of the barn find the INFANTRY.
trial: 8 1 2 Travel west and turn right at the third road. *Find the INFANTRY north of the barn.
trial: 8 1 3 Travel west and at the third road turn right. *North of the barn find the INFANTRY.

trial: 8 1 4 Viaje hacia el oeste y en el tercer camino cruce a la derecha. *Al norte del establo encuentre la INTANTERIA.

trial: 8 1 5 서쪽으로 가서 세번째 교차로에서 우회전 하라. *굿간 북쪽에있는 보병부대를 찾아라.

trial: 8 1 6 INFANTRY

trial: 8 1 7 INFANTRY

trial: 8 1 8 INFANTERIA

trial: 8 1 9 보병부대

trial: 8 1 10 icon

comment: Start at the northern most road entering from the west edge

trial: 8 2 1 Head east and at the fourth street turn south. *On the west side of road find the TANK.

trial: 8 2 2 Head east and turn south at the fourth street. *Find the TANK on the west side of road.

trial: 8 2 3 Head east and at the fourth street turn south. *On the west side of road find the TANK.

trial: 8 2 4 Dirijase hacia el este y en la cuarta calle cruce hacia el sur. *En el lado oeste del camino encuentre el TANQUE.

trial: 8 2 5 동쪽으로 따라가다가 네번째 교차로를 만나면 남쪽으로 가라. *길 서쪽에 있는 탱크를 찾아라.

trial: 8 2 6 TANK

trial: 8 2 7 TANK

trial: 8 2 8

trial: 8 2 9 탱크

trial: 8 2 10 icon

comment: Start at the northern most road entering from the west edge

trial: 8 3 1 At the first street head south. *At the first intersection turn left. *Cross the lake and locate the first GUN.

trial: 8 3 2 At the first street head south. *Turn left at the first intersection and locate the first GUN beyond the lake.

trial: 8 3 3 At the first street head south. *At the first intersection turn left. *Cross the lake and locate the first GUN.

trial: 8 3 4 En la primera calle dirijase hacia el sur. *En la primera intersección cruce hacia la izquierda. *Pase el lago y localice el primer CAÑON.

trial: 8 3 5 동쪽으로 가서 첫번째 교차로에서 남쪽으로 가라. *또 첫번째 교차로에서 좌회전 하라. *오수를 건넌후 첫번째 대포를 찾아라.
trial: 8 3 6 GUN
trial: 8 3 7 GUN
trial: 8 3 8 CAÑON
trial: 8 3 9 대포
trial: 8 3 10 icon
comment: Start at the southern most road entering from the east edge
trial: 8 4 1 Head west and cross two bridges. *At the first street turn north. *On the west side of the road locate the ANTITANK GUN.
trial: 8 4 2 Head west and turn north at the first street *following the two bridges. *Locate the ANTITANK GUN on the west side of the road.
trial: 8 4 3 Head west and cross two bridges. *At the first street turn north. *On the west side of the road locate the ANTITANK GUN.
trial: 8 4 4 Dirijase hacia el oeste y atraviese dos puentes. *En la primera calle cruce hacia el norte. *En el lado oeste del camino localice el CAÑON ANTI-TANQUE.
trial: 8 4 5 서쪽으로 가서 두번째 다리를 건넌후 첫번째 교차로에서 북쪽으로 가라. *길 서쪽에 있는 탱크격파용 대포를 찾아라.
trial: 8 4 6 ANTITANK GUN
trial: 8 4 7 ANTITANK GUN
trial: 8 4 8 CAÑON ANTI-TANQUE
trial: 8 4 9 탱크격파용 대포
trial: 8 4 10 icon
comment: Start at the southern most road entering from the east edge
trial: 8 5 1 Go west and cross the bridge. *At the first street turn right. *North of the power lines find the HOSPITAL.
trial: 8 5 2 Go west and turn right at the first street after the bridge. *Locate the HOSPITAL north of the power lines.
trial: 8 5 3 Go west and cross the bridge. *At the first street turn right. *North of the power lines find the HOSPITAL.

trial: 8 5 4 Vaya hacia el oeste y atraviese el puente. *En la primera calle cruce a la derecha. *Al norte de los cables electricos encuentre el HOSPITAL.
trial: 8 5 5 서쪽으로 가서 첫번째 다리를 건넌후 첫번째 교차로에서 *북쪽으로 가라. *송전선 북쪽에 있는 병원을 찾아라.
trial: 8 5 6 HOSPITAL
trial: 8 5 7 HOSPITAL
trial: 8 5 8 HOSPITAL
trial: 8 5 9 병원
trial: 8 5 10 icon
comment: Start at the northern most road entering from the west edge
trial: 8 6 1 Travel east. At the third street turn north. *Make the first left and on the left locate the PERSONNEL CARRIER.
trial: 8 6 2 Travel east and turn north at the third street. *Locate the PERSONNEL CARRIER on the left *after turning left on the first street.
trial: 8 6 3 Travel east. At the third street turn north. *Make the first left and on the left locate the PERSONNEL CARRIER.
trial: 8 6 4 Viaje hacia el este. En la tercera calle cruce hacia el norte. *Cruce en la primera izquierda y en el lado izquierdo localice el VEHICULO DE TRANSPORTE DE PERSONAL.
trial: 8 6 5 동쪽으로 가서 세번째 교차로에서 북쪽으로 가라. *첫번째 교차로에서 좌회전 한후 좌측에있는 전령을 찾아라.
trial: 8 6 6 PERSONNEL CARRIER
trial: 8 6 7 PERSONNEL CARRIER
trial: 8 6 8 VEHICULO DE TRANSPORTE DE PERSONAL
trial: 8 6 9 전령
trial: 8 6 10 icon
comment: Start in the middle of the island
trial: 9 1 1 Cross the east bridge and at the first street turn south. *At the second crossroad head west and on the north side *of the street locate the INFANTRY.
trial: 9 1 2 Cross the east bridge and turn south onto the first street. *Head west at the second crossroad and locate the INFANTRY *on the north side of the street.
trial: 9 1 3 Cross the east bridge and at the first street turn south. *At the second crossroad head west and on the north side *of the street locate the INFANTRY.

trial: 9 1 4 Atraviese el puente hacia el este y en la primera calle cruce hacia el sur. *En la segunda intersección diríjase hacia el oeste y en el lado norte *de la calle localice la INFANTERIA.

trial: 9 1 5 동쪽 다리를 지난후 첫번째 길에서 남쪽으로 가라. *두번째 교차로에서 서쪽으로 가다가 북쪽에있는 보병부대를 찾아라.

trial: 9 1 6 INFANTRY

trial: 9 1 7 INFANTRY

trial: 9 1 8 INFANTERIA

trial: 9 1 9 보병부대

trial: 9 1 10 icon

comment: Start at the road entering from the east edge north of the church

trial: 9 2 1 Go west to the third street. Turn left and then east. *On the left side of the road locate the TANK.

trial: 9 2 2 Go west and turn east onto and east from the third street. *Locate the TANK on the left side of the road.

trial: 9 2 3 Go west to the third street. Turn left and then east. *On the left side of the road locate the TANK.

trial: 9 2 4 Vaya hacia el oeste hasta la tercera calle. Cruce a la izquierda y luego hacia el este. *En el lado izquierdo del camino localice el TANQUE.

trial: 9 2 5 서쪽으로 따라가다가 세번째 교차로를 만나 좌회전한후 *다시 동쪽으로 돌아가라. 길 좌측에 있는 탱크를 찾아라.

trial: 9 2 6 TANK

trial: 9 2 7 TANK

trial: 9 2 8 TANQUE

trial: 9 2 9 탱크

trial: 9 2 10 icon

comment: Start at the western most road which heads directly south

trial: 9 3 1 Go south and after crossing the bridge turn east. *Continue through the crossroads and on the north side *of the street select the first GUN.

trial: 9 3 2 Go south and turn east after crossing the bridge. *Continue through the crossroads and select the first GUN *on the north side of the street.

trial: 9 3 3 Go south and after crossing the bridge turn east. *Continue through the crossroads and on the north side *of the street select the first GUN.
trial: 9 3 4 Vaya hacia el sur y después de atravesar el puente cruce hacia el este. *Pase el cruce y en el lado norte *de la calle seleccione el primer CAÑON.
trial: 9 3 5 남쪽으로 가서 첫번째 다리를 지난후 동쪽으로 가라. *계속 두 교차로들을 지나 길 북쪽에 있는 첫 대포를 찾아라.
trial: 9 3 6 GUN
trial: 9 3 7 GUN
trial: 9 3 8 CAÑON
trial: 9 3 9 대포
trial: 9 3 10 icon
comment: Start at the road entering from the east edge north of the church
trial: 9 4 1 Travel west and at the second road turn north. *Cross onto the island and to the right of the bridge *and locate the ANTITANK GUN.
trial: 9 4 2 Turn north at the second road as you travel west. *Cross onto the island and locate the ANTITANK GUN *to the right of the bridge.
trial: 9 4 3 Travel west and at the second road turn north. *Cross onto the island and to the right of the bridge *and locate the ANTITANK GUN.
trial: 9 4 4 Viaje hacia el oeste y en el segundo camino cruce hacia el norte. *Cruce hacia la isla y a la derecha del puente *localice el CAÑON ANTI-TANQUE.
trial: 9 4 5 서쪽으로 따라가다가 두번째 교차로를 만나 북쪽으로 가라. *섬으로 가는 다리를 건넌후 다리 우측에 있는 *탱크격파용 *대포를 찾아라.
trial: 9 4 6 ANTITANK GUN
trial: 9 4 7 ANTITANK GUN
trial: 9 4 8 CAÑON ANTI-TANQUE
trial: 9 4 9 탱크격파용 대포
trial: 9 4 10 icon
comment: Start at the road entering from the east edge north of the church
trial: 9 5 1 Head west to the first intersection and turn south. *Take the right fork and *after the second street indicate the HOSPITAL.

trial: 9 5 2 Head west and turn south at the first intersection. *Indicate the HOSPITAL after following the right fork *and crossing the second street.
trial: 9 5 3 Head west to the first intersection and turn south. *Take the right fork and *after the second street indicate the HOSPITAL.
trial: 9 5 4 Dirijase hacia el oeste hasta la primera intersección y cruce hacia el sur. *Tome el lado derechc de la bifurcación y *después de la segunda calle señale el HOSPITAL.
trial: 9 5 5 서쪽으로 가다가 첫번째 교차로에서 남쪽으로 가라. *갈림길에서 우회전 하여 두번째 교차로를 지난후 병원을 찾아라.
trial: 9 5 6 HOSPITAL
trial: 9 5 7 HOSPITAL
trial: 9 5 8 HOSPITAL
trial: 9 5 9 병원
trial: 9 5 10 icon
comment: Start at the western most road which heads directly south
trial: 9 6 1 Head south to the third street and turn east. *After crossing one bridge locate the PERSONNEL CARRIER.
trial: 9 6 2 Head south. Turn east on the third street *and locate the PERSONNEL CARRIER after crossing one bridge.
trial: 9 6 3 Head south to the third street and turn east. *After crossing one bridge locate the PERSONNEL CARRIER.
trial: 9 6 4 Dirijase hacia el sur hasta la tercera calle y cruce hacia el este. *Después de atravesar un puente localice EL VEHICULO DE TRANSPORTE DE PERSONAL.
trial: 9 6 5 남쪽으로 가서 세번째 교차로에서 동쪽으로 가라. *다리를 아나 건넌후 전령을 찾아라.
trial: 9 6 6 PERSONNEL CARRIER
trial: 9 6 7 PERSONNEL CARRIER
trial: 9 6 8 VEHICULO DE TRANSPORTE DE PERSONAL
trial: 9 6 9 전령
trial: 9 6 10 icon
comment: Start at northern road entering from the west edge
trial: 10 1 1 Travel east across the bridge. *Turn at the fourth street and locate the INFANTRY.

trial: 10 1 2 Travel east and turn at the fourth street *after crossing the bridge. Locate the INFANTRY.
trial: 10 1 3 Travel east across the bridge. *Turn at the fourth street and locate the INFANTRY.
trial: 10 1 4 Viaje hacia el este y atraviese el puente. *Cruce en la cuarta calle y localice la INFANTERIA.
trial: 10 1 5 동쪽으로 가다가 다리를 건너라. *북쪽으로 향한 네번째 교차로에서 우회전 하여 보병부대를 찾아라.
trial: 10 1 6 INFANTRY
trial: 10 1 7 INFANTRY
trial: 10 1 8 INFANTERIA
trial: 10 1 9 보병부대
trial: 10 1 10 icon
comment: Start at northern road entering from the west edge
trial: 10 2 1 Follow the road east and turn at the third northbound street. *On the west side of the road indicate the TANK.
trial: 10 2 2 Turn at the third northbound street encountered *while following the road to the east. *Indicate the TANK on the west side of the road.
trial: 10 2 3 Follow the road east and turn at the third northbound street. *On the west side of the road indicate the TANK.
trial: 10 2 4 Siga el camino que va hacia el este y cruce en la tercera calle que se dirije hacia el norte. *En el lado oeste del camino señale el TANQUE.
trial: 10 2 5 동쪽으로 따라가다가 세번째 북쪽으로 향한 교차로를 만나면 •북쪽으로 가라. •길 서쪽에 있는 탱크를 찾아라.
trial: 10 2 6 TANK
trial: 10 2 7 TANK
trial: 10 2 8 TANQUE
trial: 10 2 9 탱크
trial: 10 2 10 icon
comment: Start at northern road entering from the west edge
trial: 10 3 1 Head east and on the second street turn south. *After passing the church locate the first GUN.
trial: 10 3 2 Head east. Turn south on the second street *and locate the first GUN after passing the church.
trial: 10 3 3 Head east and on the second street turn south. *After passing the church locate the first GUN.

trial: 10 3 4 Dirijase hacia el este y en la segunda calle cruce hacia el sur. *Después de pasar la iglesia localice el primer CAÑON.
trial: 10 3 5 동쪽으로 가서 두번째 교차로에서 남쪽으로 가라. *그 길을 따라 교회를 지난후 첫번째 대포를 찾아라.
trial: 10 3 6 GUN
trial: 10 3 7 GUN
trial: 10 3 8 CAÑON
trial: 10 3 9 대포
trial: 10 3 10 icon
comment: Start at northern road entering from the east edge
trial: 10 4 1 Head west and at the second intersection turn south. *At the second crossroads turn right *and locate the first ANTITANK GUN.
trial: 10 4 2 Head west and then south at the second intersection. *Locate the first ANTITANK GUN after turning fight *at the second crossroads.
trial: 10 4 3 Head west and at the second intersection turn south. *At the second crossroads turn right *and locate the first ANTITANK GUN.
trial: 10 4 4 Dirijase hacia el oeste y en la segunda intersección cruce hacia el sur. *En la segunda transversal cruce hacia la derecha *y localice el primer CAÑON ANTI-TANQUE.
trial: 10 4 5 서쪽으로 가다가 두번째 교차로에서 남쪽으로 가라. *또 두번째 교차로에서 우회전 한후 첫번째 나타나는 *탱크저항용 대포를 찾아라.
trial: 10 4 6 ANTITANK GUN
trial: 10 4 7 ANTITANK GUN
trial: 10 4 8 CAÑON ANTI-TANQUE
trial: 10 4 9 탱크저항용 대포
trial: 10 4 10 icon
comment: Start at northern road entering from the west edge
trial: 10 5 1 Travel east. Cross the bridge and turn south. *At the first street head west *and at the far end of the block indicate the HOSPITAL.
trial: 10 5 2 Travel east and turn south after crossing the bridge. *Head west on the first street and indicate the HOSPITAL *at the far end of the block.
trial: 10 5 3 Travel east. Cross the bridge and turn south. *At the first street head west *and at the far end of the block indicate the HOSPITAL.

trial: 10 5 4 Viaje hacia el este. Atraviese el puente y cruce hacia el sur. *En la primera calle dirijase hacia el oeste *y al final de la cuadra señale el HOSPITAL.
trial: 10 5 5 동쪽으로 가다가 다리를 건넌후 남쪽으로 향해있는 길을따라가라. *첫번째 교차로에서 서쪽으로 가서 막다른 길에 있는 *병원을 찾아라.
trial: 10 5 6 HOSPITAL
trial: 10 5 7 HOSPITAL
trial: 10 5 8 HOSPITAL
trial: 10 5 9 병원
trial: 10 5 10 icon
comment: Start at northern road entering from the east edge
trial: 10 6 1 Travel west and turn at the second southbound street. *Turn again at the first northbound street *and near the bridge find the PERSONNEL CARRIER.
trial: 10 6 2 Travel west and turn at the second southbound street. *At the first northbound street turn again *and find the PERSONNEL CARRIER near the bridge.
trial: 10 6 3 Travel west and turn at the second southbound street. *Turn again at the first northbound street *and near the bridge find the PERSONNEL CARRIER.
trial: 10 6 4 Viaje hacia el oeste y cruce en la segunda calle que se dirije hacia el sur. *Cruce nuevamente en la primera calle que se dirije hacia el norte *y cerca del puente encuentre el VEHICULO DE TRANSPORTE DE PERSONAL.
trial: 10 6 5 서쪽으로 가서 두번째 남쪽으로 향한 길에서 남쪽으로 가라. *첫번째 북쪽으로 향한 길에서 북쪽으로 가다가 다리 *근처에있는 전령을 찾아라.
trial: 10 6 6 PERSONNEL CARRIER
trial: 10 6 7 PERSONNEL CARRIER
trial: 10 6 8 VEHICULO DE TRANSPORTE DE PERSONAL
trial: 10 6 9 전령
trial: 10 6 10 icon

APPENDIX E

EXPERIMENTAL TASK

PROGRAM SOURCE CODE

```
PROGRAM MapTask;

        USES
                MacPrint, XTTypeDefs, Extender1, MapInit, MapProcs2, MapProcs3,
MapProcs4, SubjectInfo, MapGraphics;

        VAR
                event : EventRecord;
                whatHappened : EventStuff;
                anEvent : boolean;
                appleMenu, fileMenu, editMenu : MenuHandle;
                menu3, menu4, menu5, menu6 : MenuHandle;
                BreakOut : boolean;

        PROCEDURE SetupMainMenus;
        BEGIN
                StdMenus(applemenu, filemenu, editMenu);
                SetItem(applemenu, 1, 'HEL HCI Task Set');
                EnableItem(applemenu, 1);
                menu3 := BuildMenu(23, 'Task', 'Read Scripts...;Subject Information;-;Run;-
;Save Data');
                menu4 := BuildMenu(24, 'Utilities', 'Compile Stimuli...;-;Analyze');
                menu5 := BuildMenu(25, 'Script', 'none');
                menu6 := BuildMenu(26, 'Participant', 'none');
                DisableItem(menu3, 3);
                DisableItem(menu3, 5);
                DisableItem(menu3, 4); {Run}
                DisableItem(menu3, 6); {Save Data}
                DisableItem(menu4, 2);
        END;

        FUNCTION MakeItem (VAR item : STRING) : str255;
                VAR
                        len : integer;
        BEGIN
                len := length(item);
                IF len <= 16 THEN
                        MakeItem := item
                ELSE
                        MakeItem := concat(copy(item, 1, 14), '...');
        END;

        PROCEDURE blot (WhichIcon : integer);   {to blot out target}
                VAR
                        theRect : rect;
                        x, y : integer;
                        c : char;
        BEGIN
                x := xy[WhichIcon].h;
                y := xy[WhichIcon].v;
                TextFont(150);
                TextSize(24);
                TextMode(SrcXOr);
                moveto(x, y);
```

```
{c := chr(Script[trial].target);}
            drawstring(c);
        END;

        PROCEDURE GetIcon (place : point;
                                                    VAR HitTarget :
boolean);
            VAR
                i, Zoneht, x, y, wback, hup, wforward, hdown, m, t : integer;
                arect : rect;
                icon : char;
                TargetWidth, ZoneWidth : integer;
        BEGIN
            Place.v := place.v - MapOffTop; {adjust for window position}
            Place.h := Place.h - MapOffLeft;
            ZoneWidth := 32;   {total hit zone width}
            ZoneHt := 30; {total hit zone ht}
            TextFace([]);
            TextFont(150); {could be a Param}
            TextSize(24);
            icon := chr(Targets[Script[block, trial].target]);
            TargetWidth := CharWidth(icon);
            HitTarget := false;
            wback := (ZoneWidth - TargetWidth) DIV 2; {zone extent to left of symbol
origin}
            wforward := ZoneWidth - wback;  {zone extent to right of symbol origin}
            hdown := 6;
            hup := ZoneHt - hdown;
            m := Script[block, trial].map;
            t := Script[block, trial].target;
            x := Maps[m, t].tx + IconXadj;
            y := Maps[m, t].ty + IconYadj;
            IF (place.h >= x - wback) AND (place.h < x + wforward) AND (place.v >= y -
hup) AND (place.v < y + hdown) THEN
                BEGIN
                    HitTarget := true;
                    setrect(arect, x - wback, y - hup, x + wforward, y + hdown);
                    PenSize(2, 2);
                    FrameRect(arect);
                END;
        END;


        PROCEDURE GetResponse;
            VAR
                found, timeout, break, response : boolean;
                TheEvent : EventRecord;
                stop, ticks, elapsed, now, now1, now2 : longint;
                x, y : integer;
                place : point;
                chCode : integer;
                ch : char;
                WhichIcon : integer;
                NumFound, NumErrors : integer;
```

```
                p : longint;
BEGIN
        SetPort(MapWindow);
        FlushEvents(EveryEvent, 0);
        timeout := false;
        break := false;
        response := false;
        NumFound := 0;
        NumErrors := 0;
        StartTicks := tickcount;
        REPEAT
                now := tickcount;
                now1 := now + 1;
                now2 := now + 2;
                elapsed := now - StartTicks;
                IF elapsed >= Condition[Script[block, trial].condition].length THEN
                        BEGIN
                                timeout := true;
                                sysbeep(10);
                        END;

                ticks := Condition[Script[block, trial].condition].length - elapsed;
                IF (ticks MOD 60 = 0) THEN
                        countdown(ticks);
                IF GetNextEvent(EveryEvent, TheEvent) THEN
                        BEGIN
                                stop := TheEvent.when;
                                ticks := stop - StartTicks;
                                CASE TheEvent.what OF

                                        MouseDown :
                                                BEGIN
                                                        place := TheEvent.where;
                                                        GetIcon(place, found);
                                                        IF found THEN
                                                                BEGIN
                                                                        response :=
true;
{blot(WhichIcon);}
                                                                        Script[block,
trial].RT := ticks;
                                                                        Script[block,
trial].hit := true;
                                                                        delay(30, p);
                                                                END;
                                                        IF NOT found THEN
                                                                BEGIN
                                                                        Script[block,
trial].Errors := Script[block, trial].Errors + 1;
                                                                        sysbeep(1);
                                                                END;
                                                END; {mousedown}

                                        KeyDown :
```

```
                                                BEGIN
                                                        chCode :=
BitAnd(TheEvent.message, CharCodeMask);
                                                        ch := CHR(chCode);
                                                        IF ch = '?' THEN {abort trial}
                                                                break := true;
                                                        IF (ch = '1') OR (ch = '2') THEN
                                                                SysBeep(2);
                                                        IF ch = '~' THEN {pause}
                                                                readln(ch);
                                                        IF ch = 'l' THEN
                                                                BEGIN
                                                                        BreakOut :=
true; {out of block and script}
                                                                        break := true;
{out of trial}
                                                                END;
                                                END; {keydown}
                                        OTHERWISE
                                                ;
                                END; {case}
                        END; {begin}
                UNTIL timeout OR break OR response;
        END;

        PROCEDURE Wait (start, wait : longint);
                VAR
                        elapsed : longint;
        BEGIN
                REPEAT
                        elapsed := tickcount - start;
                UNTIL elapsed >= wait;
        END;

        PROCEDURE CalculatePerformance (se : integer);
                VAR
                        n, t : integer;
        BEGIN
                Criterion.meanhits := 0;
                Criterion.meanrt := 0;
                Criterion.meanerrors := 0;
                FOR t := 1 TO ScriptEvent[se].ntrials DO
                        BEGIN
                                Criterion.Meanerrors := Criterion.Meanerrors + Script[block,
t].errors;
                                IF Script[block, t].hit = true THEN
                                        BEGIN
                                                Criterion.Meanhits := Criterion.Meanhits + 1;
                                                Criterion.MeanRT := Criterion.MeanRT +
Script[block, t].RT;
                                        END;
                        END;

                IF Criterion.Meanhits > 0 THEN
```

```
            Criterion.MeanRT := Criterion.MeanRT / Criterion.Meanhits;
        Criterion.Meanerrors := Criterion.Meanerrors / ScriptEvent[se].ntrials;
        Criterion.Meanhits := Criterion.Meanhits / ScriptEvent[se].ntrials;

        WITH Criterion DO
                BEGIN
                        hits := true;
                        errors := true;
                        RT := true;
                        IF MeanHits < Criterion.HitsCriterion THEN
                                hits := false;
                        IF MeanRT > Criterion.RTCriterion THEN
                                RT := false;
                        IF MeanErrors > Criterion.ErrorsCriterion THEN
                                errors := false;
                END;
END;

PROCEDURE ReportPerformance;
        VAR
                p : longint;
                sec : real;
                h : integer;
BEGIN
        SetPort(WarnWindow);
        ShowWindow(WarnWindow);
        TextFont(3);
        TextSize(12);
        TextFace([]);
        TextMode(SrcCopy);
        moveto(15, 18);
        Drawstring('Your performance on this practice set:');
        moveto(20, 34);
        drawstring('Targets found:  ');
        TextFace([bold]);
        IF Criterion.hits THEN
                drawstring('good ')
        ELSE
                BEGIN
                        sysbeep(4);
                        drawstring('too few ');
                END;
        h := round(Criterion.meanhits * 100);
        TextFace([]);
        WriteDraw('(', h : 3, ' %)');


        moveto(20, 50);
        TextFace([]);
        drawstring('Speed:  ');
        TextFace([bold]);
        IF Criterion.RT THEN
                drawstring('good ')
        ELSE
```

```
            BEGIN
                sysbeep(4);
                drawstring('too slow ');
            END;
        TextFace([]);
        sec := Criterion.meanRT / 60;
        WriteDraw('(', sec : 2 : 1, ' sec)');


        moveto(20, 66);
        TextFace([]);
        drawstring('Errors:   ');
        TextFace([bold]);
        IF Criterion.errors THEN
            drawstring('good ')
        ELSE
            BEGIN
                sysbeep(4);
                drawstring('too many ');
            END;
        TextFace([]);
        WriteDraw('(', Criterion.meanerrors : 2 : 1, ' per trial)');

        moveto(20, 86);
        TextFace([bold]);
        drawstring('Click trackball/mouse to continue...');
        FlushEvents(EveryEvent, 0);
        REPEAT
        UNTIL Button;
        HideWindow(WarnWindow);

END;

PROCEDURE DoMapTask;
    VAR
        t, b, i, o, se, loc : integer;
        a : real;
        p : longint;
        PicNum, PicID : integer;
        thePicHand : PicHandle;
        check : boolean;
        fbytes, gbytes : size;
        therect : rect;
        RightNow : longint;
        oldPort : GrafPtr;
BEGIN
    BreakOut := false;
    GotData := true;
    HideWindow(GraphWindow);
    GetPort(oldPort);
    MakeMapWindow;
    MessageWindow;
    MakeWarningWindow;
    SetRect(therect, 0, 20, 512, 345);
```

```
CritTrainNum := 0;

se := 0;
REPEAT
        se := se + 1;

        IF ScriptEvent[se].event = 'ibi' THEN
                BEGIN
                        loc := ScriptEvent[se].location;
                        delay(IBI[loc].length, p);
                        Warning(IBI[loc].warn, IBI[loc].message);
                END; {ibi}

        IF ScriptEvent[se].event = 'trials' THEN
                BEGIN
                        block := ScriptEvent[se].location;
                        REPEAT
                                trial := 0;
                                REPEAT
                                        trial := trial + 1;
                                        RightNow := 0; {init value for 1st trial}
                                        PicNum := Script[block, trial].map;
                                        PicID := MapIds[PicNum];
                                        MapPic := GetPicture(PicID);

                                        HLock(Handle(MapPic));
                                        PlaceIcons;
                                        MergePictures;
                                        IF NOT MergeError THEN
                                                BEGIN
                                                        Wait(RightNow,
Params.ITI);
                                                        SetPort(MapWindow);

        ShowWindow(MapWindow);

                                                        WHILE
GetNextEvent(everyevent, Event) DO  {ForceEvents}

        HandleEvent(Event, WhatHappened);
                                                        KillPicture(MergePic);
                                                        DeliverPrompt;

                                                        GetResponse;
                                                        HideWindow(Mwindow);
                                                        RightNow := tickcount;

        HideWindow(MapWindow);
                                                END; {not MergeError}

                                UNTIL (trial = ScriptEvent[se].ntrials) OR
BreakOut;

                                IF Criterion.practice AND NOT BreakOut THEN
```

```
                                                    BEGIN
                                                        CritTrainNum := CritTrainNum
+ 1;
                                                        CalculatePerformance(se);
                                                        ReportPerformance;
                                                        FOR t := 1 TO
ScriptEvent[se].ntrials DO
                                                            BEGIN
                                                                Script[block,
t].hit := false;
                                                                Script[block,
t].errors := 0;
                                                                Script[block,
t].RT := 0;
                                                            END;
                                                    END;

                                        UNTIL ((Criterion.hits AND Criterion.RT AND
Criterion.errors) OR NOT Criterion.practice OR BreakOut);

                                    END; {if trials}
                            UNTIL (se = nscript) OR BreakOut;

                            IF Criterion.practice THEN
                                    Information.trainNum := CritTrainNum;

                            SysBeep(1);
                            SysBeep(3);
                            SysBeep(5);
                            IF NOT Criterion.practice THEN
                                    SaveData;
                            KillWindow(Mwindow);
                            KillWindow(WarnWindow);
{SetWPic(MapWindow, nil);}
                            KillWindow(MapWindow);{put this in loop above:}
{ReleaseResource(Handle(thePicHand));}
{KillPicture(thePicHand);}
                            TextFont(3); {helps return menu bar to normal}
                            TextSize(12);
                            TextFace([]);
                            fbytes := MaxMem(gbytes);
                            writeln('heap size after compaction=', fbytes);
                            SetPort(oldPort);
                    END;

BEGIN
        HideAll;
 {hide all LS Pascal windows}
        XTendInit;
        Watch := GetCursor(WatchCursor);
        TitlePage;
        InitInfo; {subject info set to blank}
        SetupMainMenus;
        LoadedMapScript := false; {GotData is inited in ClearArrays}
```

```
LoadedMapParams := false;
GotInformation := false;
BreakOut := false;
MakeGraphicsWindow;
REPEAT
    REPEAT
        SystemTask;  {Enables desk accessories to work}
    UNTIL XTGetNextEvent(EveryEvent, event);
    HandleEvent(event, whatHappened);
    WITH whatHappened DO
        CASE MenuNum OF
            11 :
                BEGIN
                    IF ItemNum = 1 THEN
                        sysbeep(5);
                END;
            23 :
                BEGIN
                    IF ((ItemNum = 4) AND LoadedMapScript AND
GotInformation) THEN
                        BEGIN
                            DisableItem(applemenu, 0);
                            DisableItem(filemenu, 0);
                            DisableItem(editmenu, 0);
                            DisableItem(menu3, 0);
                            DisableItem(menu4, 0);
                            DisableItem(menu5, 0);
                            DisableItem(menu6, 0);
                            DrawMenuBar;
                            DoMapTask;
                            EnableItem(applemenu, 0);
                            EnableItem(filemenu, 0);
                            EnableItem(editmenu, 0);
                            EnableItem(menu3, 0);
                            EnableItem(menu4, 0);
                            EnableItem(menu5, 0);
                            EnableItem(menu6, 0);
                            DrawMenuBar;
                            writeln('memory at end of map
task = ', FreeMem);
                        END;
                    IF ItemNum = 1 THEN
                        BEGIN
                            OpenScript;
                            IF LoadedMapScript THEN
                                BEGIN
                                    SetItem(menu5,
1, Params.Title);
                                    drawMenuBar;
                                END;
                        END;
                    IF ItemNum = 2 THEN
                        BEGIN
                            GetSomeInformation;
```

```
                                                    IF GotInformation THEN
                                                        BEGIN
                                                            SetItem(menu6,
1, Information.name);                                       .
                                                            drawMenuBar;
                                                        END;
                                                END;
{if ItemNum = 6 then SaveData;}
                                                IF LoadedMapScript AND GotInformation THEN
{enable RUN if script is read}
                                                    EnableItem(menu3, 4);
{if LoadedMapScript and GotInformation and GotData then EnableItem ( menu3 , 6 );}
{enable RUN if script is read}
                                END; {menu 3}

                            24 :
                                BEGIN
                                    IF ItemNum = 1 THEN
                                        CompileScript;
                                    IF ItemNum = 3 THEN
                                        DoMapAnalysis;

                                END;

                            OTHERWISE
                                ;
                        END; {case}
        UNTIL (ExitRequest(WhatHappened) = TRUE);
        KillWindow(GraphWindow);
        TextFont(3); {helps return menu bar to normal}
        TextSize(12);
        TextFace([]);

END.

UNIT MapProcs2;

INTERFACE

        USES
            MacPrint, XTTypeDefs, Extender1, MapInit;

        PROCEDURE MakeMapWindow;
        PROCEDURE MessageWindow;
        PROCEDURE MakeWarningWindow;
        PROCEDURE MakeInfoWindow;
        PROCEDURE Warning (WarnLag : longint;
                                                    message : STRING);
        PROCEDURE SaveData;


IMPLEMENTATION
```

```
        PROCEDURE MakeMapWindow;
                VAR
                        myRect : rect;
                        visible, goAwayBox, growBox, vScrollBar, hScrollBar : boolean;
        BEGIN
                visible := false;
                goAwayBox := false;
                growBox := false;
                vScrollBar := false;
                hScrollBar := false;
{SetRect(myRect, -13, 9, 520, 342);}
 {L,T,R,B}
                MapOffTop := 10; {menu bar}
                MapOffLeft := -10;
                SetRect(myRect, MapOffLeft, MapOffTop, 520, 300); {L,T,R,B}
                MapWindow := CreateWindow(MapWR, myRect, '', 2, visible, goAwayBox,
growBox, vScrollBar, hScrollBar);
        END;

        PROCEDURE MessageWindow;
                VAR
                        visible, goAwayBox, growBox, vScrollBar, hScrollBar : boolean;
                        myRect : rect;
        BEGIN
                visible := false;
                goAwayBox := false;
                growBox := false;
                vScrollBar := false;
                hScrollBar := false;
                SetRect(myRect, 3, 295, 509, 340); {L,T,R,B}
                MwIndow := CreateWindow(MWR, myRect, '', 1, visible, goAwayBox,
growBox, vScrollBar, hScrollBar);
        END;

        PROCEDURE MakeWarningWindow;
                VAR
                        myRect : rect;
                        visible, goAwayBox, growBox, vScrollBar, hScrollBar : boolean;
        BEGIN
                visible := false;
                goAwayBox := false;
                growBox := false;
                vScrollBar := false;
                hScrollBar := false;
                SetRect(myRect, 100, 100, 400, 200); {L,T,R,B}
                WarnWindow := CreateWindow(WarnWR, myRect, '', 1, visible, goAwayBox,
growBox, vScrollBar, hScrollBar);
        END;

        PROCEDURE MakeInfoWindow;
                VAR
                        myRect : rect;
                        visible, goAwayBox, growBox, vScrollBar, hScrollBar : boolean;
        BEGIN
```

```
            visible := true;
            goAwayBox := false;
            growBox := false;
            vScrollBar := false;
            hScrollBar := false;
            SetRect(myRect, 75, 100, 435, 200); {L,T,R,B}
            InfoWindow := CreateWindow(InfoWR, myRect, '', 1, visible, goAwayBox,
growBox, vScrollBar, hScrollBar);
      END;

      PROCEDURE Warning;
            VAR
                  seconds : longint;
                  start, now, left : longint;
                  sw, ww, wx, sx : integer;
      BEGIN
            SetPort(WarnWindow);
            TextFont(3);
            TextSize(12);
            TextFace([bold]);
            Chop(message);
            sw := StringWidth(message);
            ww := sw + 60;
            IF ww < 200 THEN
                  ww := 200;
            wx := 256 - ww DIV 2;
            MoveWindow(Warnwindow, wx, 100, false);
            SizeWindow(Warnwindow, ww, 100, true);
            sx := (ww DIV 2) - (sw DIV 2);
            ShowWindow(WarnWindow);
            TextMode(SrcCopy);
            moveto(sx, 30);
            drawstring(message);
            seconds := WarnLag DIV 60;
            start := tickcount;
            REPEAT
                  now := tickcount;
                  left := WarnLag - (now - start);
                  seconds := left DIV 60;
                  moveto((ww DIV 2) - 40, 65);
                  writedraw(seconds, '  ');
            UNTIL left <= 0;
            HideWindow(WarnWindow);
      END;


{**********************************************************************
**************}
      PROCEDURE SaveData;
            VAR
                  b, t, i, trials, se : integer;
                  rn : longint;
                  filename, day, hour, month : str255;
                  dt : datetimerec;
```

```
BEGIN
        SetCursor(Watch^^);
        filename := copy(information.name, 1, 12);
        i := pos(':', filename); {make sure it doesn't look like a pathname}
        IF i <> 0 THEN
                filename := '';
        IF filename = '' THEN
                filename := 'Unnamed';
        GetTime(dt);
        NumToString(dt.day, day);
        NumToString(dt.hour, hour);
        NumToString(dt.month, month);
        day := copy(day, 1, 2);
        hour := copy(hour, 1, 2);
        month := copy(month, 1, 2);
        filename := concat(filename, ' ', month, ',', day, ',', hour);
        writeln('filename=', filename);
        rewrite(outfile, filename);

        WITH information DO
                BEGIN
                        writeln('writing info to file:', filename, 'name:', name);
                        writeln(outfile, name);
                        writeln(outfile, ethnic);
                        writeln(outfile, cond);
                        writeln(outfile, date);
                        writeln(outfile, trainnum);
                        writeln(outfile, comment);
                        writeln(outfile, Params.Title);
                END;
        writeln(outfile, nblock);
        FOR se := 1 TO nscript DO
                IF ScriptEvent[se].event = 'trials' THEN
                        BEGIN
                                writeln(outfile, ScriptEvent[se].ntrials);
                                b := ScriptEvent[se].location;
                                FOR t := 1 TO ScriptEvent[se].ntrials DO
                                        writeln(outfile, Script[b, t].condition, Script[b,
t].hit, Script[b, t].RT, Script[b, t].errors);
                                END;

        FOR se := 1 TO nscript DO
                BEGIN
                        IF ScriptEvent[se].event = 'trials' THEN
                                BEGIN
                                        b := ScriptEvent[se].location;
                                        FOR t := 1 TO ScriptEvent[se].ntrials DO
                                                writeln(outfile, 'trial:', Script[b,
t].map, Script[b, t].target, Script[b, t].condition);
                                        END; {trial}

                        IF ScriptEvent[se].event = 'ibi' THEN
                                BEGIN
                                        b := ScriptEvent[se].location;
```

```
                                        writeln(outfile, 'lbl', lBl[b].length, lBl[b].warn,
lBl[b].message);
                                    END; {ibl}
                
                    END; {se}
                close(outfile);
                InitCursor;
        END; {SAVE DATA}
{*****************************************************************
***************}


END.

UNIT MapProcs3;

INTERFACE

        USES
                MacPrint, XTTypeDefs, Extender1, ListManager, XTTypeDefs2, Extender2,
MapInit;

        PROCEDURE MergePictures;
        PROCEDURE TitlePage;
        PROCEDURE PlaceIcons;
        PROCEDURE DeliverPrompt;
        PROCEDURE CountDown (ticks : longint);

IMPLEMENTATION


        PROCEDURE MergePictures;
                VAR
                        offbit : BitMap;
                        err : OSErr;
                        tempPic, sillyPic : PicHandle;
                        check : boolean;
                        aPort : GrafPtr;
                        mem : longint;
                        gbytes, fbytes : size;
        BEGIN
                MergeError := false;
                mem := FreeMem;
{writeln('memory at start of merge = ', mem, ' trial:', trial);}
{fbytes := MaxMem(gbytes);}
{writeln('heap size after compaction=', fbytes);}

                tempPic := AddPic(MapPic, IconPic);
                HLock(Handle(tempPic));
                SetRect(offbit.bounds, 0, 0, 600, 400);
                err := PicToBits(tempPic, offbit);
                IF err <> noErr THEN
                        BEGIN
                                sysbeep(10);
                                writeln('OSerr=', err),
```

```
                    MergeError := true;
                END
            ELSE
                BEGIN
                    MergePic := BitsToPic(offbit);
                    HLock(Handle(MergePic));
                    KillBits(offbit);
                    KillPicture(IconPic);
                    ReleaseResource(Handle(MapPic));
                    KillPicture(MapPic);
                    KillPicture(tempPic);
                    SetPort(MapWindow);
                    SetWPic(MapWindow, MergePic);
                END;
        END;

        PROCEDURE TitlePage;
            VAR
                tempPic : PicHandle;
                aRect, myrect : Rect;
                offbit : BitMap;
                err : OSErr;
                TitleWindow : WindowPtr;
                visible, goAwayBox, growBox, vScrollBar, hScrollBar : boolean;
                TWR : WindowRecord;
                theevent : eventrecord;
        BEGIN
            tempPic := GetPicture(1000);
            HLock(Handle(tempPic));
            SetRect(offbit.bounds, 0, 0, 520, 345);
            err := PicToBits(tempPic, offbit);

            visible := true;
            goAwayBox := false;
            growBox := false;
            vScrollBar := false;
            hScrollBar := false;
            SetRect(myRect, -15, 0, 530, 350); {L,T,R,B}
            TitleWindow := CreateWindow(TWR, myRect, '', 2, visible, goAwayBox,
growBox, vScrollBar, hScrollBar);
            SetPort(TitleWindow);
            SetWPic(TitleWindow, tempPic);
{SetRect(myRect, -15, 0, 530, 350);}
 {L,T,R,B}
{DrawPicture(tempPic, myRect);}

            ForceEvents(everyEvent);
            FlushEvents(EveryEvent, 0);
            REPEAT
            UNTIL Button;
{until GetNextEvent(everyevent, theevent);}
            KillWindow(TitleWindow);
            ReleaseResource(Handle(tempPic));
            KillPicture(tempPic);
```

```
END;

PROCEDURE PutIcon (ascii, x, y : integer);
    VAR
        icon, mask : char;
        Xoff, Yoff : integer;
BEGIN
    icon := chr(ascii);
    mask := chr(ascii + 1);
    moveto(x + IconXadj, y + IconYadj);
    TextMode(SrcBic);
    drawstring(mask);   {draw mask with BIC}
    moveto(x + IconXadj, y + IconYadj);
    TextMode(SrcOr);
    drawstring(icon);   {draw icon with OR}
END;

PROCEDURE PlaceIcons;
    VAR
        i, j, x, y, m, t, ascii, theTarget : integer;
        picrect : rect;
BEGIN
    SetRect(picrect, 0, 0, 520, 345);
    IconPic := OpenPicture(picrect);
    TextFont(150);
    TextSize(24);
    TextFace([]);
    m := Script[block, trial].map;
    theTarget := Script[block, trial].target; {the icon that is the target this time}


{first place hot target symbol}
    x := Maps[m, theTarget].tx; {position of the target}
    y := Maps[m, theTarget].ty;
    ascii := Targets[theTarget];
    PutIcon(ascii, x, y);

{put down START and 2 target distractors if not a single-target condition}
    IF NOT Condition[Script[block, trial].condition].single THEN
        BEGIN
            x := Maps[m, theTarget].sx; {position of the start}
            y := Maps[m, theTarget].sy;
            ascii := 162; {needs a mask}
            PutIcon(ascii, x, y);

            x := Maps[m, theTarget].d1x; {position of the dist}
            y := Maps[m, theTarget].d1y;
            ascii := Targets[theTarget];
            PutIcon(ascii, x, y);

            x := Maps[m, theTarget].d2x; {position of the dist}
            y := Maps[m, theTarget].d2y;
            ascii := Targets[theTarget];
            PutIcon(ascii, x, y);
```

```
                END; {if not icon}

{place the non-target icons}
            FOR t := 1 TO MaxTargets DO
                IF t <> theTarget THEN
                    BEGIN
                        x := Maps[m, t].tx;
                        y := Maps[m, t].ty;
                        ascii := Targets[t];
                        PutIcon(ascii, x, y);

                        x := Maps[m, t].d1x;
                        y := Maps[m, t].d1y;
                        ascii := Targets[t];
                        PutIcon(ascii, x, y);

                        x := Maps[m, t].d2x;
                        y := Maps[m, t].d2y;
                        ascii := Targets[t];
                        PutIcon(ascii, x, y);
                    END; {if not t}
            ClosePicture;
            HLock(Handle(IconPic));
        END;

        PROCEDURE CountDown;
            VAR
                seconds : longint;
        BEGIN
            seconds := ticks DIV 60;
            SetPort(MWindow);
            TextMode(SrcCopy);
            TextFont(3);
            TextSize(12);
            TextFace([bold]);
            moveto(480, 15);
            writedraw(seconds : 3, '  ');
            SetPort(MapWindow);
        END;

        PROCEDURE Parse (thePrompt : str255;
                                                    VAR prompts :
StrArray128;
                                                    VAR lines : integer;
                                                    VAR max : integer);
            VAR
                p, i : integer;
                finished : boolean;
        BEGIN
            FOR i := 1 TO 5 DO
                prompts[i] := ' ';
            finished := false;
            i := 0;
            lines := 0;
```

```
            REPEAT
                    i := i + 1;
                    p := pos('*', thePrompt);
                    IF p = 0 THEN
                            BEGIN
                                    prompts[i] := thePrompt; {no parsing in prompt}
                                    lines := lines + 1;
                                    finished := true;
                            END
                    ELSE
                            BEGIN
                                    prompts[i] := copy(thePrompt, 1, p - 1);
                                    delete(thePrompt, 1, p);
                                    lines := lines + 1;
                            END;
            UNTIL finished;
{next part assumes that Text size and face have been set correctly}
            max := 0;
            FOR i := 1 TO lines DO
                    IF StringWidth(prompts[i]) > max THEN
                            max := StringWidth(prompts[i]);
    END;

    PROCEDURE DeliverPrompt;
            VAR
                    thePrompt : str255;
                    prompts : StrArray128;
                    theRect : Rect;
                    c : char;
                    mem : longint;
                    font : integer;
                    yoff, i, lines, max, m, t : integer;
    BEGIN
            m := Script[block, trial].map;
            t := Script[block, trial].target;
            IF Script[block, trial].message <> 'icon' THEN
                    BEGIN
                            font := Condition[Script[block, trial].condition].font;
                            SetPort(Mwindow);
                            TextMode(SrcOr);
                            TextFont(font);
                            TextSize(Params.PromptSize);
                            TextFace([]);
                            thePrompt := Script[block, trial].message;
                            Parse(thePrompt, prompts, lines, max);
                            yoff := Params.PromptSize + 3;
                            SelectWindow(Mwindow);
                            ShowWindow(Mwindow);
                            IF lines > 1 THEN
                                    BEGIN
                                            FOR i := 1 TO lines DO
                                                    IF prompts[i] <> '' THEN
                                                            BEGIN
```

```
                                                    moveto(10, yoff + 3 +
(l - 1) * yoff);

        DrawString(prompts[l]);
                                                              END;
                                    END; {multiple lines}

                                IF lines = 1 THEN
                                    BEGIN
                                        i := StringWidth(prompts[1]);
                                        moveto((512 - i) DIV 2, 25);
                                        DrawString(prompts[1]);
                                    END;
                        END; {word}

                IF Script[block, trial].message = 'icon' THEN
                        BEGIN
                            SetPort(Mwindow);
                            TextMode(SrcOr);
                            font := Condition[Script[block, trial].condition].font;
                            TextFont(font);
                            TextSize(24);
                            TextFace([]);
                            SelectWindow(Mwindow);
                            ShowWindow(Mwindow);
                            c := chr(targets[Script[block, trial].target]);
                            moveto(245, 25);
                            drawstring(c);
                        END; {icon}

                TextFont(3);
                TextSize(9);
                TextFace([]);
                moveto(470, 30);
                writedraw(Script[block, trial].map : 2, Script[block, trial].target : 2,
Script[block, trial].condition : 3);
        END;
END.

UNIT MapProcs4;

INTERFACE

        USES
                MacPrint, XTTypeDefs, Extender1, MapInit;

        PROCEDURE OpenScript;
        PROCEDURE CompileScript;

IMPLEMENTATION

        CONST
                MaxScan = 21; {number of items allowed in a line of the script}
        TYPE
```

```
        wordarray = ARRAY[1..MaxScan] OF STRING[MaxScriptLength];
        intarray = ARRAY[1..MaxScan] OF integer;
        longarray = ARRAY[1..MaxScan] OF longint;
        StimMapRec = RECORD
                        pos : integer;
                        selected : boolean;
                        available : boolean;
                END;
        StimMapArray = ARRAY[1..MaxMaps, 1..MaxTargets, 1..MaxConditions] OF
StimMapRec;

        StimListRec = RECORD
                        map : integer;
                        target : integer;
                        cond : integer;
                        selected : boolean;
                END;
        StimListArray = ARRAY[1..MaxBlocks, 1..MaxTrialsPerBlock] OF StimListRec;

        StimRec = RECORD
                        map : integer;
                        target : integer;
                        cond : integer;
                        prompt : STRING[MaxScriptLength];
                END;
        StimFile = FILE OF StimRec;
        TrialSetArray = ARRAY[0..MaxConditions, 1..MaxScript] OF integer;

VAR
        ncond, nmaps : integer;
        nstimuli : integer; {# of items in stimuli section of script}
        words : wordarray;
        int : intarray;
        long : longarray;
        inString : str255;
        instruction : STRING[18];
        OneStim : StimRec;
        RandFile : StimFile;
        randfilename : str255;
        posit : integer;
        DWR : WindowRecord;
        Dwindow : WindowPtr;

PROCEDURE DialogWindow;
        VAR
                visible, goAwayBox, growBox, vScrollBar, hScrollBar : boolean;
                myRect : rect;
BEGIN
        visible := true;
        goAwayBox := false;
        growBox := false;
        vScrollBar := false;
        hScrollBar := false;
        SetRect(myRect, 82, 255, 429, 330); {L,T,R,B}
```

```
        Dwindow := CreateWindow(DWF., myRect, '', 1, visible, goAwayBox, growBox,
vScrollBar, hScrollBar);
    END;

    PROCEDURE GiveInfo (line : integer;
                                                    msg : STRING);
        VAR
            r : rect;
            x, y : integer;
    BEGIN
        SetPort(Dwindow);
        x := 20;
        y := 15 + line * 18;
        SetRect(r, 0, y - 15, 350, 75);
        EraseRect(r);
        TextFont(3);
        TextSize(12);
        TextFace([]);
        moveto(x, y);
        drawstring(msg);
    END;


    PROCEDURE CountTrials (VAR TrialSet : TrialSetArray); {count # of trials used in each
block; # may vary}
        VAR
            b, s, c, n : integer;
    BEGIN
        FOR b := 1 TO nblock DO
            BEGIN
                n := 0;
                BEGIN
                    FOR c := 1 TO MaxConditions DO (# of trials in block b)
                        n := n + TrialSet[c, b];
                    TrialSet[0, b] := n;
                    FOR s := 1 TO MaxScript DO
                        IF ScriptEvent[s].location = b THEN (find place
in ScriptEvent to put n)
                            ScriptEvent[s].ntrials := n;  {working
from block# back to script#}
                END;
            END;
    END;

    PROCEDURE SelectStimuli (eventsequence : integer;
                                                    StimMap :
StimMapArray;
                                                    TrialSet :
TrialSetArray;
                                                    VAR StimList :
StimListArray);
{theLocation is the blocks dimension of the TrialSet and Script arrays}
{TrialSet contains the number of trials to run of each condition in the}
{block indicated by theLocation. 1st dim of TrialSet is condition#}
```

```
        VAR
                i, t, try, sequence, cond, wanted, theLocation : integer;
                trymap, trytarget, trylist, listlength : integer;
                found : boolean;
    BEGIN
{for each condition, randomly choose the number needed, update the}
{.selected field, and make a list of the stim #s.  Store list with nscript}
        theLocation := ScriptEvent[eventsequence].location;
        listlength := ScriptEvent[eventsequence].ntrials;
{writeln('working on theLocation ', theLocation : 1, ' listlength=', listlength : 2);}
        FOR cond := 1 TO ncond DO
            BEGIN
{writeln('processing condition ', cond : 2, ' Number to find:', TrialSet[cond, theLocation]);}
                FOR t := 1 TO TrialSet[cond, theLocation] DO {how many of
cond t to find}
                    BEGIN
                        found := false;
                        WHILE NOT (found) DO
                            BEGIN
                                trymap := Randomize(nmaps);
                                trytarget :=
Randomize(MaxTargets);
{writeln('trymap=', trymap : 2, 'trytarget=', trytarget : 2);}
                                IF StimMap[trymap, trytarget,
cond].available AND NOT StimMap[trymap, trytarget, cond].selected THEN
                                    BEGIN
                                        found := true;

    StimMap[trymap, trytarget, cond].selected := true;
                                        found := false;
                                        WHILE NOT
(found) DO {randomize order within blocks}
                                            BEGIN

    trylist := randomize(listlength);

    IF StimList[theLocation, trylist].selected = false THEN

        found := true;
                                            END;
{writeln(' t=', t : 2, ' found map # ', trymap : 2, ' target: ', trytarget : 1, ' trylist:', trylist :
2);}

    StimList[theLocation, trylist].map := trymap;

    StimList[theLocation, trylist].target := trytarget;

    StimList[theLocation, trylist].cond := cond;

    StimList[theLocation, trylist].selected := true;
                                    END; {found}
                            END; {while}
                    END;{for t}
            END; {for cond}
```

```
END; {proc}

PROCEDURE StrToIntLong (theWords : wordarray;
                                                    N : integer);
    VAR
        I : integer;
        neww : str255;
        ii : Longint;
BEGIN
    FOR i := 1 TO N DO
        BEGIN
            neww := words[i];
            IF ((theWords[i] <> '') AND (length(neww) < 12)) THEN
                BEGIN
                    StringToNum(theWords[i], ii);
                    long[i] := ii;
                    int[i] := LoWord(ii),
                END;
        END;
END;

PROCEDURE ScanScript
    VAR
        len, p, w, first, last : integer;
        i : longint;
        tab : char;
        tabs : ARRAY[0..MaxScan] OF integer;
        wd : str255;
BEGIN
    tab := chr(9);
    FOR i := 1 TO MaxScan DO
        BEGIN
            words[i] := '';
            int[i] := 0;
            long[i] := 0;
            tabs[i] := 0;
        END;
    tabs[0] := 0;
    len := length(instring);
    p := 0;
    FOR i := 1 TO len DO
        BEGIN
            IF instring[i] = tab THEN
                BEGIN
                    p := p + 1;
                    tabs[p] := i;
                END;
        END;
    FOR i := 1 TO p DO
        BEGIN
            first := tabs[i - 1] + 1;
            last := tabs[i] - 1;
            wd := copy(instring, first, last - first + 1);
            IF length(wd) <= MaxScriptLength THEN
```

```
                                        words[i] := wd
                                    ELSE
                                        BEGIN
                                            writeln('instring=', instring, 'wd=', wd,
'length=', length(wd));
                                            sysbeep(10);
                                            sysbeep(10);
                                            sysbeep(10);
                                        END;
                        END;
                    first := tabs[p] + 1; {pick up last word in line}
                    last := len;
                    wd := copy(instring, first, last - first + 1);
                    IF length(wd) <= MaxScriptLength THEN
                        words[i] := wd
                    ELSE
                        BEGIN
                            writeln('instring=', instring, 'wd=', wd, 'length=', length(wd));
                            sysbeep(10);
                            sysbeep(10);
                            sysbeep(10);
                        END;
{StrToIntLong(words, MaxScan);}

{       writeln(instring);}
{    for i := 1 to p + 1 do}
{writeln(i,  words[i]);}
        END;


        FUNCTION UpperCase (instr : STRING) : STRING;
            VAR
                i, len, a : integer;
                c : char;
        BEGIN
            len := length(instr);
            FOR i := 1 TO len DO
                BEGIN
                    c := instr[i];
                    a := ord(c);
                    IF ((a >= 97) AND (a <= 122)) THEN
                        BEGIN
                            a := a - 32;
                            c := chr(a);
                            instr := omit(instr, i, 1);
                            insert(c, instr, i);
                        END;
                END;
            UpperCase := instr;
        END;

        PROCEDURE ClearArrays (VAR StimList : StimListArray;
                                                    VAR StimMap :
StimMapArray;
```

```
                                                    VAR TrialSet :
TrialSetArray);
        VAR
                i, j, k, m, t, c : integer;
    BEGIN
        FOR i := 1 TO MaxBlocks DO
                FOR j := 1 TO MaxTrialsPerBlock DO
                        BEGIN
                                Script[i, j].condition := 0;
                                Script[i, j].map := 0;
                                Script[i, j].target := 0;
                                Script[i, j].message := '';
                                Script[i, j].RT := 0;
                                Script[i, j].errors := 0;
                                Script[i, j].hit := false;
                        END;

        WITH Params DO
                BEGIN
                        ITI := 0;
                        StartPoints := 0;
                        DivPoints := 0;
                        ModPoints := 0;
                        PromptSize := 0;
                        Title := 'Untitled';
                END;

        FOR i := 1 TO MaxScript DO
                BEGIN
                        ScriptEvent[i].event := '';
                        ScriptEvent[i].location := 0;
                        FOR j := 1 TO MaxConditions DO
                                TrialSet[j, i] := 0;
                END;
        FOR i := 1 TO MaxTrialsPerBlock DO
                FOR j := 1 TO MaxBlocks DO
                        BEGIN
                                StimList[j, i].map := 0;
                                StimList[j, i].cond := 0;
                                StimList[j, i].target := 0;
                                StimList[j, i].selected := false;
                        END;
        FOR m := 1 TO MaxMaps DO
                FOR t := 1 TO MaxTargets DO
                        FOR c := 1 TO MaxConditions DO
                                BEGIN
                                        StimMap[m, t, c].pos := 0;
                                        StimMap[m, t, c].selected := false;
                                END;

        Criterion.practice := false;
        Criterion.HitsCriterion := 0;
        Criterion.ErrorsCriterion := 0;
        Criterion.RTCriterion := 0;
```

```
        GotData := false;
    END;

    FUNCTION GetInstruction (s : STRING) : STRING;
        VAR
            p : integer;
    BEGIN
        p := pos(':', s);
        getinstruction := UpperCase(copy(s, 1, p));
    END;

    PROCEDURE ReadParams (VAR TrialSet : TrialSetArray);
        VAR
            i, j, m, t, c, s, n : integer;
            nibi : integer;
    BEGIN
        ncond := 0;
        nscript := 0;
        nmaps := 0;
        nibi := 0;
        nblock := 0;

        reset(infile, filename);

        WHILE NOT eof(infile) DO
            BEGIN
                readln(infile, inString);
                IF instring <> '' THEN
                    BEGIN
                        instruction := GetInstruction(inString);
                        IF ((instruction <> 'COMMENT:') AND
(instruction <> 'TRIAL:')) THEN
                            BEGIN
                                GiveInfo(2, inString);
                                ScanScript;
                                StrToIntLong(words,
MaxScan);

                                IF instruction = 'TITLE:' THEN
                                    Params.Title :=
words[2];

                                IF instruction = 'ITI:' THEN
                                    Params.ITI := long[2];

                                IF instruction = 'PROMPTSIZE:'
THEN
                                    Params.PromptSize :=
int[2];

                                IF instruction = 'POINTS:' THEN
                                    BEGIN
```

```
        Params.StartPoints := int[2];
        Params.DivPoints := int[3];
        Params.ModPoints := int[4];
                                                        END;
                                                IF instruction = 'TARGETS:'
THEN
                                                        FOR i := 1 TO
MaxTargets DO
                                                                Targets[i] :=
int[i + 1];
                                                IF instruction = 'TRAINING:'
THEN
                                                        BEGIN
                                                                IF words[2] =
'yes' THEN
        Criterion.practice := true;
        Criterion.HitsCriterion := int[3] / 100;
        Criterion.ErrorsCriterion := int[4] / 100;
        Criterion.RTCriterion := long[5];
                                                        END;
                                                IF instruction = 'MAPIDS:' THEN
                                                        BEGIN
                                                                FOR i := 1 TO
MaxMaps DO
                                                                        BEGIN
        j := i * 2;
        IF int[j] > 0 THEN
                BEGIN
                        nmaps := nmaps + 1;  {used in random selection}
                        Mapids[int[j]] := int[j + 1];
                END;
                                                                        END;
                                                        END; {MAPS}
                                                IF instruction = 'COND:' THEN
                                                        BEGIN
                                                                ncond := ncond
+ 1;
```

```
        Condition[int[2]].font := int[3];
        Condition[int[2]].length := long[4];
                                                    IF words[5] =
'single' THEN
        Condition[int[2]].single := true;
                                                        END;
                                                IF instruction = 'IBI:' THEN
                                                    BEGIN
                                                        nscript :=
nscript + 1;
                                                        nibi := nibi + 1;
        ScriptEvent[nscript].event := 'ibi';
        ScriptEvent[nscript].location := nibi;
                                                        IBI[nibi].length
:= long[2];
                                                        IBI[nibi].warn
:= long[3];
        IBI[nibi].message := words[4];
                                                        END;
                                                IF instruction = 'BLOCK:' THEN
                                                    BEGIN
                                                        nscript :=
nscript + 1;
                                                        nblock := nblock
+ 1;
        ScriptEvent[nscript].event := 'trials';
        ScriptEvent[nscript].location := nblock;
                                                        FOR i := 1 TO
MaxConditions DO
                                                            BEGIN
        j := i * 2;
        IF int[j] > 0 THEN
            TrialSet[int[j], nblock] := int[j + 1];
                                                            END;
                                                    END; {trials}
                                                IF instruction = 'MAP:' THEN
                                                    BEGIN
                                                        m := int[2];
                                                        t := int[3];
```

```
                                                                WITH Maps[m,
t] DO
                                                                    BEGIN
        sx := int[4]; {position of the target}

        sy := int[5];

        tx := int[6]; {position of the start symbol}

        ty := int[7];

        d1x := int[8];

        d1y := int[9]; {position of the distractor sysmbols}

        d2x := int[10];

        d2y := int[11];
                                                                        END;
{with}
                                                                  END; {map}
                                                          END; {not trial or comment}
                                                END;{not a blank}
                        END;{while}
                close(infile);
                LoadedMapParams := true;
        END; {ReadParams}

        PROCEDURE ReadCompiledScript (VAR StimMap : StimMapArray);
                VAR
                        i, n, m, t, c : integer;

        BEGIN
                FOR m := 1 TO MaxMaps DO
                        FOR t := 1 TO MaxTargets DO
                                FOR c := 1 TO MaxConditions DO
                                        StimMap[m, t, c].pos := 0;

                open(randfile, randfilename);
                posit := 0;
                WHILE NOT eof(randfile) DO
                        BEGIN
                                seek(randFile, posit);
                                posit := filepos(randfile);
                                OneStim := randFile^;
                                WITH OneStim DO
                                        BEGIN
                                                IF ((cond > MaxConditions) OR (map > MaxMaps)
OR (target > MaxTargets)) THEN
                                                        BEGIN
                                                                writeln(map, target, cond);
                                                                cond := 1;
                                                                map := 1;
```

```
                                                target := 1;
                                                sysbeep(10);
                                                sysbeep(10);
                                                sysbeep(10);
                                        END;
                                StimMap[map, target, cond].pos := posit;
                                StimMap[map, target, cond].available := true;
                        END;
                posit := posit + 1;
        END; {while}

    close(randfile);

    LoadedMapScript := true;
END; {Proc}

PROCEDURE CompileScript;
    VAR
        outfilename, s : str255;
        i, n, m, t, c, pos : integer;
BEGIN
    filename := OldFileName('Raw Script file:');
    IF filename <> '' THEN
        BEGIN
            outfilename := NewFileName('Processed file:');
            IF outfilename <> '' THEN
                BEGIN
                    SetCursor(Watch^^);
                    reset(infile, filename);
                    open(randfile, outfilename);

                    pos := 0;
                    WHILE NOT eof(infile) DO
                        BEGIN

                            FOR i := 1 TO MaxScan DO
                                BEGIN
                                    words[i] := '';
                                    int[i] := 0;
                                    long[i] := 0;
                                END;

                            readln(infile, inString);
                            IF instring <> '' THEN
                                BEGIN
                                    instruction :=
GetInstruction(inString);
                                    IF instruction =
'TRIAL:' THEN
                                        BEGIN

    ScanScript;

    StrToIntLong(words, MaxScan);
```

```
        WITH OneStim DO
            BEGIN
                map := int[2];
                target := int[3];
                cond := int[4];
                prompt := words[5];
            END;
        RandFile^ := OneStim;
        put(RandFile);
                                                                    END; {if
trial}
                                                                END; {not blank}
                                                            END; {while}
                                                    close(randfile);
                                                    InitCursor;
                                                END; {OK outfile}
                                            close(infile);
                                        END; {OK infile}
        END;
        PROCEDURE LoadScript (StimList : StimListArray;
                                                                TrialSet :
TrialSetArray;
                                                                StimMap :
StimMapArray);
            VAR
                i, b, tr, n, m, c, targ, cond, t : integer;
                stimSection : boolean;
        BEGIN
            open(randfile, randfilename);
            FOR b := 1 TO nblock DO
                FOR tr := 1 TO TrialSet[0, b] DO    {# of trials in block b}
                    FOR cond := 1 TO ncond DO       {ncond is from CONDITIONS
param}
                        BEGIN
                            m := StimList[b, tr].map;
                            t := StimList[b, tr].target;
                            c := StimList[b, tr].cond;
                            posit := StimMap[m, t, c].pos;
                            seek(randFile, posit);
                            OneStim := randFile^;
                            WITH Script[b, tr] DO
                                BEGIN
                                    condition := c;
```

```
                                                    map := m;
                                                    target := t;
                                                    message := OneStim.prompt;
                                            END;
                                    END; {for}
                close(randfile);
        END;

        PROCEDURE PrintScript;
                VAR
                        s, b, t : integer;
        BEGIN
                FOR s := 1 TO nscript DO
                        BEGIN
                                writeln(ScriptEvent[s].event);
                                IF ScriptEvent[s].event = 'ibi' THEN
                                        BEGIN
                                                b := ScriptEvent[s].location;
                                                write(IBI[b].length, IBI[b].warn,
IBI[b].message);
                                                writeln;
                                        END; {ibi}
                                IF ScriptEvent[s].event = 'trials' THEN
                                        BEGIN
                                                b := ScriptEvent[s].location;
                                                FOR t := 1 TO ScriptEvent[s].ntrials DO
                                                        BEGIN
                                                                write('cond=', Script[b,
t].condition : 2, ' map=', Script[b, t].map : 2, ' target=', Script[b, t].target : 1, ' ', Script[b,
t].message);
                                                                writeln;
                                                        END; { for t}
                                        END; {trials}
                        END; {for s}
        END; {proc}

        PROCEDURE OpenScript;
                VAR
                        i, j : integer;
                        msg : STRING[128];
                        TrialSet : TrialSetArray;
                        StimList : StimListArray;
                        StimMap : StimMapArray;

        BEGIN
                seed1 := tickcount MOD 5;
                seed2 := tickcount MOD 4;
{Verify;}
                DialogWindow;
                GiveInfo(1, 'Select Parameters file...');
                GiveInfo(2, '(Ethnic group, training, etc.)');
                filename := OldFileName('Script file:');
                IF filename <> '' THEN
                        BEGIN
```

```
                    GiveInfo(1, 'Select Compiled Prompts file...');
                    randfilename := OldFileName('Script file.');
                    IF randfilename <> '' THEN

                        BEGIN
                            SetCursor(Watch^^);
                            GiveInfo(1, 'Clearing data structures.');
                            ClearArrays(StimList, StimMap, TrialSet);

                            GiveInfo(1, 'Reading parameters file.');
                            ReadParams(TrialSet);

                            GiveInfo(1, 'Reading prompts file.');
                            GiveInfo(2, '(First pass)');
                            ReadCompiledScript(StimMap);

                            GiveInfo(1, 'Selecting and rand  nizing.');
                            CountTrials(TrialSet);
                            FOR i := 1 TO nscript DO
                                IF ScriptEvent[i].event = 'trials' THEN
                                    SelectStimuli(i, StimMap,
TrialSet, StimList);

                            GiveInfo(1, 'Reading prompts file.');
                            GiveInfo(2, '(Second pass)');
                            LoadScript(StimList, TrialSet, StimMap);
{PrintScript;}

                            InitCursor;
                        END; {ok filename}
                    END; {ok randfilename}
                KillWindow(DWindow);
            END; {Script procedure}

END. {unit}

UNIT SubjectInfo;

INTERFACE

    USES
        MacPrint, XTTypeDefs  Extender1, MapInit;

    PROCEDURE GetSomeInformation;
    PROCEDURE InitInfo;

IMPLEMENTATION

    CONST
        DialogID = 12345;
        iName = 3;
        iDate = 6;
        iEthnic = 7;
        iCondition = 10;
```

```
        iComment = 12;
        iCancel = 2;
        iOK = 1;
VAR
        DP : DialogPtr;
        fRefNum : integer;


PROCEDURE InitInfo;
        VAR
                sec : longint;
                d, t : str255;
                da, ti : STRING[14];
BEGIN
        GetDateTime(sec);
        IUDateString(sec, ShortDate, d);
        IUTimeString(sec, false, t);
        da := copy(d, 1, 14);
        ti := copy(t, 1, 14);
        WITH Information DO
                BEGIN
                        name := '';
                        date := concat(da, ' ', ti);
                        ethnic := '';
                        cond := '';
                        comment := '';
                        trainnum := 0;
                END;
END;

PROCEDURE PutUpDialog;
        VAR
                itemType : integer;
                itemH : handle;
                itemRect : rect;
                date : str255;
BEGIN
        date := Information.date;
        DP := GetNewDialog(DialogID, NIL, pointer(-1));
        GetDItem(DP, iDate, itemType, itemH, itemRect);
        SetIText(itemH, Date);
        ShowWindow(DP);
END;

PROCEDURE ExtractInfo;
        VAR
                itemType : integer;
                itemH : handle;
                itemRect : rect;
                name, date, ethnic, cond, comment : str255;
BEGIN
        GetDItem(DP, iName, itemType, itemH, itemRect);
        GetIText(itemH, Name);
        GetDItem(DP, iDate, itemType, itemH, itemRect);
```

```
            GetIText(itemH, Date);
            GetDItem(DP, iEthnic, itemType, itemH, itemRect);
            GetIText(itemH, Ethnic);
            GetDItem(DP, iCondition, itemType, itemH, itemRect);
            GetIText(itemH, Cond);
            GetDItem(DP, iComment, itemType, itemH, itemRect);
            GetIText(itemH, Information.Comment);

            information.name := copy(name, 1, 32);
            information.date := copy(date, 1, 32);
            information.ethnic := copy(ethnic, 1, 32);
            information.cond := copy(cond, 1, 32);

            WITH information DO
                BEGIN
                    IF name = '' THEN
                        name := 'no name';
                    IF date = '' THEN
                        date := 'no date';
                    IF ethnic = '' THEN
                        ethnic := 'no ethnic';
                    IF cond = '' THEN
                        cond := 'no cond';
                    IF comment = '' THEN
                        comment := 'no comment';
                END;
        END;

        PROCEDURE GetSomeInformation;
            VAR
                itemNum : integer;
        BEGIN
            PutUpDialog;
            REPEAT
                ModalDialog(NIL, itemNum);
            UNTIL (itemNum = iOK) OR (itemNum = iCancel);
            IF itemNum = iOK THEN
                BEGIN
                    ExtractInfo;
                    GotInformation := true;
                END;
            DisposDialog(DP);
        END;

END.

UNIT MapGraphics;

INTERFACE

    USES
        MacPrint, XTTypeDefs, Extender1, ListManager, XTTypeDefs2, Extender2,
MapInit;
```

```
        PROCEDURE DoMapAnalysis;
        PROCEDURE MakeGraphicsWindow;

IMPLEMENTATION

        CONST
                Xaxis = 270;
                Yaxis = 10;
        TYPE
                MeansRec = RECORD
                                hits : real;
                                errors : real;
                                RT : real;
                                n : integer;
                                used : boolean;
                        END;
        VAR
                GraphPic : PicHandle;
                theRect : rect;
                trials, nb, nt : integer;
                t, i, yoff, n, c : integer;
                x : ARRAY[1..MaxConditions, 1..3] OF integer;
                Xinc : integer;
                infoX, infoY : integer;
                means : ARRAY[0..MaxConditions] OF MeansRec;
                Points : real;
                OneOff, TwoOff, ThreeOff : integer;

        PROCEDURE MakeGraphicsWindow;
                VAR
                        myRect : rect;
                        visible, goAwayBox, growBox, vScrollBar, hScrollBar : boolean;
        BEGIN
                visible := false;
                goAwayBox := false;
                growBox := false;
                vScrollBar := false;
                hScrollBar := false;
                SetRect(myRect, 0, 0, 520, 345); {L,T,R,B}
                GraphWindow := CreateWindow(GraphWR, myRect, '', 4, visible, goAwayBox,
growBox, vScrollBar, hScrollBar);
        END;

        PROCEDURE ReadtheData;
                VAR
                        i, t, se, b, c, nused, n : integer;
        BEGIN
                reset(infile, filename);
                WITH information DO
                        BEGIN
                                readln(infile, name);
                                readln(infile, ethnic);
                                readln(infile, cond);
                                readln(infile, date);
```

```
                readln(infile, trainnum);
                readln(infile, comment);
                readln(infile, Params.Title);
        END;

    readln(infile, nb);
    FOR b := 1 TO nb DO
        BEGIN
            readln(infile, nt);
            FOR t := 1 TO nt DO
                readln(infile, Script[b, t].condition, Script[b, t].hit,
Script[b, t].RT, Script[b, t].errors);
        END;
    close(infile);

    FOR i := 0 TO MaxConditions DO
        BEGIN
            Means[i].n := 0;
            Means[i].RT := 0;
            Means[i].errors := 0;
            Means[i].hits := 0;
            Means[i].used := false;
        END;
    FOR b := 1 TO nb DO
        FOR t := 1 TO nt DO
            BEGIN
                c := Script[b, t].condition;
                Means[c].used := true;
                Means[c].n := Means[c].n + 1;
                Means[c].errors := Means[c].errors + Script[b,
t].errors;
                IF Script[b, t].hit = true THEN
                    BEGIN
                        Means[c].hits := Means[c].hits + 1;
                        Means[c].RT := Means[c].RT +
Script[b, t].RT;
                    END;
            END;
    FOR c := 1 TO MaxConditions DO
        BEGIN
            IF Means[c].hits > 0 THEN
                Means[c].RT := Means[c].RT / Means[c].hits;
            IF Means[c].n > 0 THEN
                Means[c].errors := Means[c].errors / Means[c].n;
            IF Means[c].n > 0 THEN
                Means[c].hits := Means[c].hits / Means[c].n;
        END; {c}

{get grand means}
    nused := 0;
    FOR c := 1 TO MaxConditions DO
        IF Means[c].used = true THEN
            nused := nused + 1;
```

```
            FOR c := 1 TO MaxConditions DO
                BEGIN
                    Means[0].hits := Means[0].hits + Means[c].hits;
                    Means[0].errors := Means[0].errors + Means[c].errors;
                    Means[0].RT := Means[0].RT + Means[c].RT;
                END;
            Means[0].hits := Means[0].hits / nused;
            Means[0].errors := Means[0].errors / nused;
            Means[0].RT := Means[0].RT / nused;

{calculate quality points}
            Points := (Means[0].hits * 2 - Means[0].errors) * (Means[0].RT / 540);
            writeln(Means[0].hits, Means[0].errors, Means[0].RT, points);

    END; {proc}

    PROCEDURE GraphData;
        VAR
            w, h, i, j : integer;
            bar : rect;
            sec : real;
    BEGIN

{tick marks and labels on x-axis}
            PenSize(2, 2);
            PenPat(black);
            moveto(Yaxis, Xaxis);
            lineto(495, Xaxis);
            OneOff := Yaxis + 10;
            TwoOff := Yaxis + (500 - Yaxis) DIV 3;
            ThreeOff := TwoOff + (500 - Yaxis) DIV 3;
            Xinc := (TwoOff - OneOff - 10) DIV MaxConditions;
            FOR i := 1 TO MaxConditions DO
                BEGIN
                    x[i, 1] := Yaxis + OneOff + ((i - 1) * Xinc);
                    x[i, 2] := Yaxis + TwoOff + ((i - 1) * Xinc);
                    x[i, 3] := Yaxis + ThreeOff + ((i - 1) * Xinc);
                END;
            TextFont(3);
            TextSize(9);
            TextFace([]);
            FOR i := 1 TO MaxConditions DO
                FOR j := 1 TO 3 DO
                    BEGIN
                        moveto(x[i, j], Xaxis + 12);
                        writedraw(i : 2);
                    END;

            i := MaxConditions DIV 2;
            TextSize(12);
            TextFace([bold]);
            moveto(x[i, 1] - 15, Xaxis + 24);
            DrawString('Hits');
            moveto(x[i, 2] - 15, Xaxis + 24);
```

```
                DrawString('Errors');
                moveto(x[i, 3] - 15, Xaxis + 24);
                DrawString('Time');


                TextSize(9);
                TextFace([]);
                w := 10;
                FOR i := 1 TO Maxconditions DO
                        IF Means[i].used = true THEN
                                BEGIN
                                        h := Xaxis - round(Means[i].hits * 70 + 1);
                                        SetRect(bar, x[i, 1], h, x[i, 1] + 10, Xaxis);{L,T,R,B}
                                        PaintRect(bar);
                                        moveto(x[i, 1], h - 2);
                                        WriteDraw(Means[i].hits : 2 : 1);

                                        h := Xaxis - round(Means[i].errors * 70 + 1);
                                        SetRect(bar, x[i, 2], h, x[i, 2] + 10, Xaxis);{L,T,R,B}
                                        PaintRect(bar);
                                        moveto(x[i, 2], h - 2);
                                        WriteDraw(Means[i].errors : 2 : 1);

                                        h := Xaxis - round(Means[i].RT / 10 + 1);
                                        SetRect(bar, x[i, 3], h, x[i, 3] + 10, Xaxis);{L,T,R,B}
                                        PaintRect(bar);
                                        moveto(x[i, 3], h - 2);
                                        sec := Means[i].RT / 60;
                                        WriteDraw(sec : 2 : 1);
                                END
                        ELSE
                                FOR j := 1 TO 3 DO
                                        BEGIN
                                                moveto(x[i, j] + 2, Xaxis - 1);
                                                DrawString('x');
                                        END;

{-------------------------------------------------------------------------}

{subject information}
                TextMode(SrcCopy);
                TextFont(3);
                TextSize(9);
                TextFace([]);
                InfoX := Yaxis;
                InfoY := 320;
                WITH information DO
                        BEGIN
                                moveto(InfoX, InfoY);
                                WriteDraw('Name: ', name);
                                WriteDraw('  Ethnic group: ', ethnic);
                                WriteDraw('  Condition: ', cond);
                                WriteDraw('  Date: ', date);
                                WriteDraw('  TN:', trainnum : 3);
```

```
                END;
            moveto(InfoX, InfoY + 15);
            WriteDraw('Parameter file: ', Params.Title);

{quality po
            TextFont(3);
            TextSize(12);
            TextFace([bold]);
            moveto(20, 60);
            WriteDraw('Points = ', points : 5 : 1);


{-------------------------------------------------------------------------}

            ClosePicture;
            SetPort(GraphWindow);
            SetWPic(GraphWindow, GraphPic);

        END; {prccedure}


        PROCEDURE DoMapAnalysis;
        BEGIN
            filename := OldFileName('Primary data file...');
            IF filename <> '' THEN
                BEGIN
                    SetCursor(Watch^^);
                    SetPort(GraphWindow);
                    IF ValidHandle(Handle(GetWPic(GraphWindow))) THEN
                        KillPicture(GetWPic(GraphWindow));
                    SetRect(therect, 0, 0, 520, 345);
                    GraphPic := OpenPicture(therect);
                    ReadtheData;
                    GraphData;
                    SetPort(GraphWindow);
                    ShowWindow(GraphWindow);
                    InitCursor;
                END;
        END;
{=========================================================================
=======}

END.
```